



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI *ENTERPRISE RESOURCE PLANNING* PADA MODUL *HUMAN RESOURCE MANAGEMENT* YANG BERORIENTASIKAN *MULTITENANCY* DENGAN *DISTRIBUTED DATABASE*

Dwi Al Aji Suseno
NRP 5112100185

Dosen Pembimbing I
Dwi Sunaryono, S.Kom, M.Kom.

Dosen Pembimbing II
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**DESIGN AND IMPLEMENTATION OF
ENTERPRISE RESOURCE PLANNING ON HUMAN
RESOURCE MANAGEMENT MODULE WITH
MULTITENANCY ORIENTED USING
DISTRIBUTED DATABASE**

**DWI AL AJI SUSENO
NRP 5112100185**

**Supervisor I
Dwi Sunaryono, S.Kom, M.Kom.**

**Supervisor II
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI ENTERPRISE RESOURCE PLANNING PADA MODUL HUMAN RESOURCE MANAGEMENT YANG BERORIENTASIKAN MULTITENANCY DENGAN DISTRIBUTED DATABASE

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

DWI AL AJI SUSENO

NRP : 5112 100 185

Disetujui oleh Dosen Pembimbing Tugas Akhir

Dwi Sunaryono, S.Kom, M.Kom

NIP: 19720528 199702 1001

(pembimbing 1)

Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc.

NIP: 19590803 198601 1001

Ph.D. JURUSAN
TEKNIK INFORMATIKA

(pembimbing 2)

SURABAYA

JULI, 2016

[Halaman ini sengaja dikosongkan]

**RANCANG BANGUN APLIKASI *ENTERPRISE*
RESOURCE PLANNING PADA MODUL HUMAN
RESOURCE MANAGEMENT YANG
BERORIENTASIKAN *MULTITENANCY* DENGAN
*DISTRIBUTED DATABASE***

Nama Mahasiswa : Dwi Al Aji Suseno
NRP : 5112 100 185
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Dwi Sunaryono, S.Kom, M.Kom.
Dosen Pembimbing 2 : Prof. Drs. Ec. Ir. Riyanarto Sarno,
M.Sc.,Ph.D.

ABSTRAK

Tugas akhir ini merancang dan membangun sebuah aplikasi Enterprise Resource Planning (ERP) pada modul Human Resource Management (HRM). Aplikasi ini berorientasikan multitency dengan basis data terdistribusi. Modul HRM adalah modul yang menangani masalah sumber daya manusia yang ada di dalam sebuah perusahaan. Modul ini dipecah lagi menjadi beberapa submodul antara lain: Recruitment and Hiring, Assignment, Target a Goal, Payroll, dan Decruitment. Kelima modul ini dibuat untuk memenuhi proses bisnis utama dari modul HRM yakni Hired-To-Fired. Aplikasi ERP ini dibangun berdasarkan aspek multitency dengan basis data terdistribusi. Hal ini diperlukan untuk mengatasi permasalahan pada pengembangan sebelumnya.

Perancangan aplikasi ERP dibuat secara modular agar mendukung pembuatan modul HRM. Selain itu juga mendukung pengembangan atau penambahan modul di masa datang. Kemudian untuk multitency, aspek ini dibuat menggunakan MySql Cluster sebagai pendukung aspek replikasi dan High-Availability.

Aplikasi ERP khususnya HRM membantu perusahaan dalam mengatur sumber daya manusia. Hal ini terbukti dalam uji coba yang dijalankan di kelas Sistem Enterprise dimana setiap perusahaan dapat mengatur sumber daya manusia dalam jumlah yang diinginkan perusahaan. Dalam ujicoba, kami menggunakan dua business plan untuk dua perusahaan. Perusahaan B memiliki jumlah pegawai sebanyak 152 dan Perusahaan A memiliki rata-rata jumlah pegawai adalah 62 pegawai meliputi direct labour dan indirect labour. Modul HRM juga dapat menjalankan fungsi sesuai proses bisnis yang telah dibuat di awal. Mulai fungsi perekrutan pegawai sampai pemutusan hak kepegawaian. Modul HRM juga dapat terintegrasi dengan modul lain. Selain itu modul ini dapat diimplementasikan pada aspek multitenancy.

Kata kunci: *ERP (Enterprise Resource Planning), HRM (Human Resource Management), Multitenancy*

DESIGN AND IMPLEMENTATION OF ENTERPRISE RESOURCE PLANNING ON HUMAN RESOURCE MANAGEMENT MODULE WITH MULTITENANCY ORIENTED USING DISTRIBUTED DATABASE

Student Name : Dwi Al Aji Suseno
Student ID : 5112 100 185
Major : Informatics Department FTIf-ITS
Supervisor I : Dwi Sunaryono, S.Kom, M.Kom.
Supervisor II : Prof. Drs. Ec. Ir. Rryanarto Sarno, M.Sc.,Ph.D.

ABSTRACT

The final project is to design and implementation an application of Enterprise Resource Planning (ERP) module Human Resource Management (HRM). This application is multitenancy-oriented with distributed database. HRM Module is a module that handles human resources issues that exist in a company. This module is subdivided into several submodules include: Recruitment and Hiring, Assignment, Target a Goal, Payroll, and Decruitment. All module is designed to meet the key business processes of the HRM module Hired-To-Fired. ERP application is built based on aspects of the distributed database multitenancy. It is necessary to solve the problems on the previous development.

The design of ERP application is created modularly to support HRM module integrated with other module. It also supports the development or additional modules in the future. Then for multitenancy, this aspect created using MySQL Cluster as the supporting aspects of replication and high-availability.

HRM in particular ERP applications help companies in managing human resources. This was proven in trials run in class Enterprise System in which every company can manage human resources in the desired amount of companies. In tests, we used two

business plans for the two companies . Company B has a number of as many as 152 employees and Company A has an average number of employees is 62 employees includes direct labor and indirect labor. HRM module can also perform the function corresponding business processes that have been made in the beginning. Start function recruitment to termination of employment. HRM module can also be integrated with other modules. Besides these modules can be implemented on multitenancy aspect.

Keywords: *ERP (Enterprise Resource Planning), HRM (Human Resource Management), Multitenancy*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul:

RANCANG BANGUN APLIKASI *ENTERPRISE RESOURCE PLANNING* PADA MODUL *HUMAN RESOURCE MANAGEMENT* YANG BERORIENTASIKAN *MULTITENANCY* DENGAN *DISTRIBUTED DATABASE*

Melalui lembar ini, penulis ingin menyampaikan ucapan terimakasih dan penghormatan yang sebesar-besarnya kepada:

1. Bapak, Ibu, kakak dan keluarga besar yang selalu memberikan dukungan penuh untuk menyelesaikan Tugas Akhir ini.
2. Bapak Riyanarto Sarno beserta keluarga dan Bapak Dwi Sunaryono beserta keluarga selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama proses pengerjaan tugas akhir ini.
3. Bapak dan Ibu dosen Jurusan Teknik Informatika ITS yang telah banyak menyampaikan ilmu dan bimbingan yang tak ternilai harganya bagi penulis.
4. Seluruh Staf dan karyawan FTIf ITS yang banyak memberikan banyak kelancaran administrasi akademik kepada penulis.
5. Akbar, Vicky, Arya, Ashari, Nicko, Yusi, Melinda dan Afina yang telah menemani dan membantu pengerjaan Tugas Akhir ini.
6. Teman-teman Thinker Studio: Yunan, Fika, Alief, Reva dan Mr Wawang yang telah memberikan dukungan untuk pengerjaan Tugas Akhir ini.
7. Teman-teman Laboratorium Manajemen Informasi sekaligus teman-teman seperjuangan anak didik Tugas

Akhir Prof. Riyanarto yaitu Akbar, Vicky, Arya, Ashari, Nicko. Yusi, Melinda, Afina.

8. Teman-teman Administrator Laboratorium Manajemen Informasi yaitu Naufal Fakhri, Riska Adhita, Miftah, Kania, Anne, Nadia, Nanang dan Lino.
9. Teman-teman angkatan 2012 jurusan Teknik Informatika ITS yang telah menemani dan mendukung penulis selama kuliah.
10. Bapak Mujib sebagai penjaga parkir jurusan Teknik Informatika yang telah menjaga sepeda motor penulis selama pengerjaan Tugas Akhir ini.
11. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu per satu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan yang penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juli 2016

Dwi Al Aji Suseno

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
DAFTAR KODE SUMBER	xxv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan.....	2
1.3 Batasan Permasalahan	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
1.6 Metodologi	4
1.7 Sistematika Penulisan.....	5
BAB II DASAR TEORI.....	7
2.1 Penelitian Terkait	7
2.2 ERP (<i>Enterprise Resource Planning</i>).....	8
2.3 Proses Bisnis <i>Human Resource Management</i>	9
2.3.1 <i>Recruitment and Hiring</i>	9
2.3.2 <i>Assignment</i>	10
2.3.3 <i>Training</i>	10
2.3.4 <i>Target A Goal</i>	10
2.3.5 <i>Payroll</i>	10
2.3.6 <i>Decruitment</i>	10
2.4 <i>Multitenant Data Architecture</i> atau <i>Multitenancy</i>	10
2.4.1 <i>Separated Database, Separated Schema</i>	11
2.4.2 <i>Shared Database, Separated Schema</i>	11

2.4.3	<i>Shared Database, Shared Schema</i>	12
2.5	<i>Human Resource Management (HRM)</i>	13
2.6	KPI (Key Performance Indicator).....	13
2.7	Kriteria SMART.....	13
2.7.1	<i>Specific</i>	14
2.7.2	<i>Measurable</i>	14
2.7.3	<i>Achievable</i>	14
2.7.4	<i>Relevant</i>	14
2.7.5	<i>Time-bound</i>	14
2.8	Pajak Penghasilan Pph 21	15
2.9	<i>Distributed Database</i>	16
2.9.1	Replikasi	16
2.9.2	Fragmentasi	16
2.10	<i>MySQL Cluster</i>	16
2.11	PHP (<i>Hypertext Preprocessor</i>).....	17
2.12	Kerangka Kerja Yii2.....	17
BAB III ANALISIS DAN PERANCANGAN SISTEM.....		19
3.1	Analisis	19
3.1.1	Proses Bisnis.....	19
3.1.2	Analisa Data	29
3.2	Deskripsi Umum Sistem	30
3.3	Identifikasi Pengguna	31
3.4	Arsitektur Sistem	31
3.5	Spesifikasi Kebutuhan Perangkat Lunak	32
3.6	Kasus Penggunaan.....	32
3.6.1	Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan.....	37
3.6.2	Kasus Penggunaan Mengelola Pembuatan Struktur Kepegawaian	43
3.6.3	Kasus Penggunaan Mengelola Perekrutan Pegawai	46

3.6.4	Kasus Penggunaan Mengelola Penempatan Pegawai	51
3.6.5	Kasus Penggunaan Mengelola <i>Training</i>	57
3.6.6	Kasus Penggunaan Mengelola Bisnis Goal	60
3.6.7	Kasus Penggunaan Mengelola Penggajian Pegawai	65
3.6.8	Kasus Penggunaan Mengelola Pengurangan Pegawai	70
3.6.9	Kasus Penggunaan Melihat Halaman Dashboard	72
3.6.10	Kasus Penggunaan Melihat Laporan Pegawai	73
3.6.11	Kasus Penggunaan Melihat Data <i>Training</i>	74
3.6.12	Kasus Penggunaan Melihat Data Pegawai	75
3.7	Perancangan Sistem	77
3.7.1	Perancangan <i>Multitenancy</i>	77
3.7.2	Perancangan RBAC	78
3.7.3	Perancangan Tampilan	79
BAB IV IMPLEMENTASI SISTEM		83
4.1	Lingkungan Pengembangan Sistem	83
4.2	Implementasi Basis Data Terdistribusi	83
4.2.1	Instalasi Data dan SQL node pada node1 dan node2	83
4.2.2	Pemasangan Node Management pada node03	85
4.2.3	Konfigurasi Manajemen Node	86
4.2.4	Konfigurasi Data dan SQL Node	87
4.2.5	Memulai MySQL Cluster	88
4.3	Implementasi RBAC (Role Based Access Control)	90
4.3.1	Membuat Tabel Pengguna	90

4.3.2	Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna	91
4.3.3	Membuat Modul Admin	93
4.3.4	Membuat Model Tabel Autentifikasi dan Tabel Pengguna di Modul Admin	94
4.3.5	Membuat <i>Controller</i> dan <i>View</i> tabel pengguna di Modul Admin	94
4.3.6	Menambahkan Kode pada UserController Modul Admin	94
4.4	Implementasi <i>Multitenancy</i>	99
4.4.1	Menambahkan Database Untuk Tenant Baru	99
4.4.2	Login Tenant	99
4.5	Implementasi Modul <i>Human Resource Management</i>	100
4.5.1	Implementasi Antarmuka	100
4.5.2	Implementasi Program	106
BAB V PENGUJIAN DAN EVALUASI		125
5.1	Lingkungan Pengujian	125
5.2	<i>Business Plan</i>	125
5.3	Skenario Pengujian	130
5.3.1	Pengujian Fungsionalitas Sistem	130
5.3.2	Pengujian Fitur Basis Data Terdistribusi	146
5.3.3	Pengujian Fitur Multitenancy	151
5.3.4	Pengujian Fitur Mengelola Role Based Access Controll (RBAC)	152
5.4	Evaluasi Pengujian	156
5.4.1	Evaluasi Pengujian Fungsionalitas	156
BAB VI KESIMPULAN DAN SARAN		159
6.1	Kesimpulan	159
6.2	Saran	160
DAFTAR PUSTAKA		161

LAMPIRAN A – PROSES BISNIS	163
INDEKS	169
BIODATA PENULIS.....	171

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 <i>Separated Schema, Separated Database</i>	11
Gambar 2.2 <i>Shared Database, Separated Schema</i>	12
Gambar 2.3 <i>Shared Schema, Shared Database</i>	12
Gambar 2.4 <i>Skema Key Performance Indicator</i>	13
Gambar 3.1 <i>Workflow ERP Level 0</i>	24
Gambar 3.2 <i>Workflow ERP Level 1 pada modul HRM</i>	25
Gambar 3.3 <i>Recruiting</i>	26
Gambar 3.4 <i>Assigning</i>	27
Gambar 3.5 <i>Training</i>	28
Gambar 3.6 <i>Targeting</i>	28
Gambar 3.7 <i>Paying Roll</i>	29
Gambar 3.8 <i>Decruiting</i>	29
Gambar 3.9 <i>Arsitektur Sistem</i>	31
Gambar 3.10 <i>Diagram Kasus Penggunaan</i>	34
Gambar 3.11 <i>Physical Data Model</i>	35
Gambar 3.12 <i>Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan Bagian (1)</i>	40
Gambar 3.13 <i>Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan Bagian (2)</i>	41
Gambar 3.14 <i>Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan Bagian (3)</i>	42
Gambar 3.15 <i>Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Kepegawaian Bagian (1)</i>	44
Gambar 3.16 <i>Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Kepegawaian Bagian (2)</i>	45
Gambar 3.17 <i>Diagram Aktivitas Kasus Penggunaan Mengelola Perekrutan Pegawai Bagian (1)</i>	48
Gambar 3.18 <i>Diagram Aktivitas Kasus Penggunaan Perekrutan Pegawai Bagian (2)</i>	49
Gambar 3.19 <i>Diagram Aktivitas Kasus Penggunaan Perekrutan Pegawai Bagian (3)</i>	50

Gambar 3.20 Diagram Aktivitas Kasus Penggunaan Perekrutan Pegawai Bagian (4).....	51
Gambar 3.21 Diagram Aktivitas Kasus Penggunaan Penempatan Pegawai Bagian (1).....	54
Gambar 3.22 Diagram Aktivitas Kasus Penggunaan Penempatan Pegawai Bagian (2).....	55
Gambar 3.23 Diagram Aktivitas Kasus Penggunaan Penempatan Pegawai Bagian (3).....	56
Gambar 3.24 Diagram Aktivitas Kasus Penggunaan Penempatan Pegawai Bagian (4).....	57
Gambar 3.25 Diagram Aktivitas Kasus Penggunaan Mengelola Training	59
Gambar 3.26 Diagram Aktivitas Kasus Penggunaan Mengelola Bisnis Goal Bagian (1)	62
Gambar 3.27 Diagram Aktivitas Kasus Penggunaan Mengelola Bisnis Goal Bagian (2)	63
Gambar 3.28 Diagram Aktivitas Kasus Penggunaan Mengelola Bisnis Goal Bagian (3)	64
Gambar 3.29 Diagram Aktivitas Kasus Penggunaan Mengelola Penggajian Pegawai Bagian (1).....	67
Gambar 3.30 Diagram Aktivitas Kasus Penggunaan Mengelola Penggajian Pegawai Bagian (2).....	68
Gambar 3.31 Diagram Aktivitas Kasus Penggunaan Mengelola Penggajian Pegawai Bagian (3).....	69
Gambar 3.32 Diagram Aktivitas Kasus Penggunaan Mengelola Pengurangan Pegawai.....	71
Gambar 3.33 Diagram Aktivitas Kasus Penggunaan Melihat Halaman Dashboard	72
Gambar 3.34 Diagram Aktivitas Kasus Penggunaan Melihat Laporan Pegawai	74
Gambar 3.35 Diagram Aktivitas Kasus Penggunaan Melihat Data <i>Training</i>	75
Gambar 3.36 Diagram Aktivitas Kasus Penggunaan Melihat Data Pegawai	76
Gambar 3.37 Perancangan <i>Multitenancy</i>	77

Gambar 3.38. Rancang antarmuka login	78
Gambar 3.39. Rancang Antarmuka Menambahkan Pengguna Baru	79
Gambar 3.40 Rancangan Antarmuka Halaman <i>Dashboard</i>	79
Gambar 3.41 Rancangan Antarmuka Halaman Utama Submodul	80
Gambar 3.42 Rancangan Antarmuka Halaman Form	81
Gambar 3.43 Rancangan Antarmuka Halaman <i>View</i> Submodul	82
Gambar 4.1 Memulai Proses Manajemen Node	88
Gambar 4.2 Ndbd dan Proses Mysql Server dapat Dimulai	89
Gambar 4.3 Mysql pada Data Node Aktif	89
Gambar 4.4 Data Node Saling Terkoneksi	90
Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil	92
Gambar 4.6. Proses Migrasi Tabel Pengguna Berhasil	93
Gambar 4.7 Antarmuka Halaman <i>Dashboard</i>	101
Gambar 4.8 Antarmuka Halaman Job Requisition	101
Gambar 4.9 Antarmuka Halaman Candidate	102
Gambar 4.10 Antarmuka Halaman <i>Shift</i>	102
Gambar 4.11 Antarmuka Halaman <i>Employee</i>	103
Gambar 4.12 Antarmuka Halaman <i>Key Performance Indicator</i>	104
Gambar 4.13 Antarmuka Halaman <i>Attendance List</i>	104
Gambar 4.14 Antarmuka Halaman <i>Payroll</i>	105
Gambar 4.15 Antarmuka Halaman <i>Decruitment</i>	105
Gambar 5.1. Pengujian fitur replikasi pada sistem	146
Gambar 5.2. Pengujian fitur replikasi pada database server1	147
Gambar 5.3. Pengujian fitur replikasi pada database server 2... ..	147
Gambar 5.4. Pengujian Fitur <i>High-Availability</i> pada Sistem	148
Gambar 5.5. Pengecekan Availability pada Node Server	148
Gambar 5.6. Pengecekan Database pada Node 1	149
Gambar 5.7. Pengecekan availability pada node server	149
Gambar 5.8 Mematikan Database Server pada node 2	150
Gambar 5.9 Proses <i>High Availability</i> (Sinkronisasi)	150
Gambar 5.10 Hasil dari Fitur <i>High Availability</i>	151
Gambar 5.11 Hasil Pengujian <i>Multitenancy</i>	152

Gambar 5.12 Proses Login User Admin.....	154
Gambar 5.13. Tampilan awal setelah login berhasil dilakukan.	154
Gambar 5.14. Admin menambahkan user baru	155
Gambar 5.15. Proses user admin menyunting data user.....	155
Gambar 5.16. Proses user admin menghapus data user.....	156
Gambar A.1 Workflow <i>Make To Stock</i> Level 1	163
Gambar A.2 Workflow <i>Make To Order</i> Level 1	164
Gambar A.3 Workflow InoERP	165
Gambar A.4 Workflow Adempiere	166
Gambar A.5 Workflow Odoo	167

DAFTAR TABEL

Tabel 2.1 Perbedaan Modul ERP 2013 dengan 2016.....	7
Tabel 2.2 Sub Modul HRM.....	8
Tabel 2.3 Penghasilan Netto Kena Pajak	15
Tabel 2.4. PTKP Kelompok Pertama	15
Tabel 2.5. PTKP Kelompok Kedua.....	15
Tabel 3.1 Kekurangan dan Kelebihan pada Odoo, Adempiere, dan InoERP	20
Tabel 3.2 Nama Modul Level 0.....	25
Tabel 3.3 Nama Submodul Level 1	26
Tabel 3.4 Daftar Kebutuhan Fungsional Sistem.....	32
Tabel 3.5 Keterangan Kode Kasus Penggunaan.....	37
Tabel 3.6 Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan.....	38
Tabel 3.7 Kasus Penggunaan Mengelola Pembuatan Struktur Kepegawaian	43
Tabel 3.8 Kasus Penggunaan Mengelola Perekrutan Pegawai....	46
Tabel 3.9 Kasus Penggunaan Mengelola Penempatan Pegawai..	52
Tabel 3.10 Kasus Penggunaan Mengelola <i>Training</i>	57
Tabel 3.11 Kasus Penggunaan Mengelola Bisnis Goal.....	60
Tabel 3.12 Kasus Penggunaan Mengelola Penggajian Pegawai .	65
Tabel 3.13 Kasus Penggunaan Mengelola Pengurangan Pegawai	70
Tabel 3.14 Kasus Penggunaan Melihat Halaman Dashboard.....	72
Tabel 3.15 Kasus Penggunaan Melihat Laporan Pegawai	73
Tabel 3.16 Kasus Penggunaan Melihat Data <i>Training</i>	74
Tabel 3.17 Kasus Penggunaan Melihat Data Pegawai	75
Tabel 5.1. Daftar Barang Jadi.....	126
Tabel 5.2. Daftar Aset	126
Tabel 5.3 Daftar Bahan Baku	127
Tabel 5.4. Daftar material.....	128
Tabel 5.5. Perencanaan keuangan perusahaan	129
Tabel 5.6 Pengujian Mengelola Struktur Perusahaan.....	130
Tabel 5.7 Pengujian Mengelola Struktur Kepegawaian	132

Tabel 5.8 Pengujian Mengelola Perekrutan Pegawai 134

Tabel 5.9 Pengujian Mengelola Penempatan Pegawai 135

Tabel 5.10 Pengujian Mengelola Training 137

Tabel 5.11 Pengujian Mengelola Bisnis Goal 138

Tabel 5.12 Pengujian Mengelola Penggajian Pegawai 140

Tabel 5.13 Pengujian Mengelola Pengurangan Pegawai 142

Tabel 5.14 Pengujian Melihat Halaman Dashboard 143

Tabel 5.15 Pengujian Melihat Laporan Pegawai 144

Tabel 5.16 Pengujian Melihat Data Training 145

Tabel 5.17 Pengujian Melihat Data Pegawai 145

Tabel 5.18 Pengujian Fitur *Multitenancy* 151

Tabel 5.19. Pengujian fitur Role Based Access Control (RBAC)
..... 152

Tabel 5.20 Evaluasi Pengujian Fungsionalitas 156

DAFTAR KODE SUMBER

Kode Sumber 4.1 Membuat grup MySQL pengguna baru dan menambah user MySQL.....	84
Kode Sumber 4.2. Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, mengubah arsip dan menciptakan symlink ke dalam direktori MySql.....	84
Kode Sumber 4.3. Mengubah lokasi ke direktori mysql.	84
Kode Sumber 4.4. Mengatur izin yang diperlukan oleh server MSQL.....	85
Kode Sumber 4.5. Menyalin script startup MySQL ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi	85
Kode Sumber 4.6. Mengubah lokasi ke dalam direktori / var / tmp direktori, mengekstrak ndb_mgm dan ndb_mgmd dari arsip ke direktori yang sesuai seperti / usr / local / bin.	86
Kode Sumber 4.7. Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi.....	86
Kode Sumber 4.8. Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri.	86
Kode Sumber 4.9. Mengatur file “config.ini”	87
Kode Sumber 4.10. Data dan SQL Node	87
Kode Sumber 4.11 Memulai Proses Manajemen Node.....	88
Kode Sumber 4.12 Memulai Proses ndbd dan proses mysql server	88
Kode Sumber 4.13 Memulai proses ndbd dan proses mysql server.	89
Kode Sumber 4.14 Generate Tabel RBAC.....	91
Kode Sumber 4.15 Kode Fungsi Tabel AuthItem	98
Kode Sumber 4.16 Kode Fungsi Tabel ItemChild	99
Kode Sumber 4.17 Kode Fungsi Tabel AuthAssignment	99
Kode Sumber 4.18 Penambahan Database Tenant Baru	99
Kode Sumber 4.19 Login Tenant	100
Kode Sumber 4.20 Kode Sumber Halaman <i>Dashboard</i>	108

Kode Sumber 4.21 Kode Sumber Melihat *Job Requisition*.....108

Kode Sumber 4.22 Kode Sumber Menambah *Job Requisition* .109

Kode Sumber 4.23 Kode Sumber Menyunting *Job Requisition*110

Kode Sumber 4.24 Kode Sumber Menghapus *Job Requisition* 110

Kode Sumber 4.25 Kode Sumber Rejected Candidate111

Kode Sumber 4.26 Kode Sumber Menyunting *Shift*112

Kode Sumber 4.27 Kode Sumber Generate *Attendance List*.....114

Kode Sumber 4.28 Kode Sumber *Generate Payroll*122

Kode Sumber 4.29 Kode Sumber Menghapus *Payroll*123

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Human Resource Management (HRM) adalah fungsi dalam sebuah organisasi atau perusahaan untuk mengatur masalah perekrutan, pengaturan dan pengarahan pekerjaan agar tercapai sebuah bisnis goal yang ingin dicapai oleh perusahaan. HRM juga membantu sebuah perusahaan untuk menyelesaikan masalah yang terjadi di dalam lingkup sumber daya manusia. Jika sebuah perusahaan tidak memiliki manajemen sumber daya manusia, maka perusahaan tersebut sulit sekali untuk berkembang dan menyelesaikan masalah yang terjadi di dalam lingkup sumber daya manusianya. Maka dari itu, HRM menjadi fungsi di dalam perusahaan yang berpengaruh besar terhadap perkembangan perusahaan.

HRM menghadapi tantangan bagaimana mengatasi masalah yang berhubungan dengan perilaku manusia dengan cara yang sangat logis atau bisa diterima oleh akal. Solusi untuk mengatasi masalah tersebut dengan fungsi-fungsi: *Recruitment and Hiring, Training, Compensation and Benefits Management* dan *Payroll*. Fungsi-fungsi ini bertujuan untuk mencapai bisnis gol perusahaan. Fungsi *Recruitment* merupakan proses awal untuk merekrut pegawai agar dapat memenuhi kebutuhan perusahaan dalam mencapai bisnis goal perusahaan. Berawal dari pembukaan lowongan pekerjaan sampai tahap *hiring*. Di tahap *hiring* inilah perusahaan menambah pegawai. Pada proses ini pula keputusan diambil oleh sebuah perusahaan berdasarkan kebutuhan yang telah dideskripsikan sebelumnya untuk mencapai *business goal* perusahaan. Lalu, fungsi *Traning* merupakan fungsi peningkatan

mutu kerja sekaligus pengembangan untuk Sumber Daya Manusia (SDM) yang ada di perusahaan. Banyak perusahaan menengahkan ke bawah yang tidak mementingkan proses ini. Padahal fungsi ini sangat berdampak besar terhadap peningkatan mutu produk maupun perusahaan. Fungsi *Target A Goal* merupakan fungsi pemberian hak terhadap kewajiban yang telah dikerjakan oleh pegawai perusahaan. Fungsi ini bertujuan untuk penentuan bonus pegawai berdasarkan kinerja yang telah dicapai oleh SDM. Fungsi *Payroll* merupakan fungsi untuk mengatur gaji para pegawai yang ada di perusahaan secara jelas dan rinci.

Pada ERP2013 lalu telah dibuat sebuah aplikasi ERP. Aplikasi ini menggunakan arsitektur berorientasi *service* (SOA) dengan sistem *Model-View-Controller* (MVC) dan *Workflow* untuk '.NET'. Namun pada aplikasi ERP ini masih belum ada modul yang menangani sumber daya manusia.. Dengan penambahan modul HRM pada rancang bangun sistem ERP 2016 diharapkan dapat mengoptimalkan kinerja perusahaan. Selain itu modul HRM dibangun berorientasi *multitenancy* dengan basis data terdistribusi.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini antara lain:

1. Bagaimana membuat proses bisnis dari modul HRM yang terdiri dari submodul-submodul *Recruitment and Hiring*, *Assignment*, *Target A Goal*, *Payroll*, dan *Decruitment* ?
2. Bagaimana membuat modul HRM (*Human Resource Management*) di dalam aplikasi ERP yang terintegrasi dengan modul lain untuk perusahaan?
3. Bagaimana mengimplementasikan multitenansi pada modul HRM?
4. Bagaimana mengatasi kelemahan *separated database* pada pengembangan sebelumnya dalam hal *high-availability* dan replikasi?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, antara lain:

1. Hasil dari tugas akhir ini adalah sebuah aplikasi ERP modul HRM yang terdiri dari submodul-submodul *Recruitment and Hiring*, *Assignment*, *Target A Goal*, *Payroll*, dan *Decruitment*.
2. Fungsional Modul:
 - a. Fungsi *Recruitment and Hiring*
 - b. Fungsi *Assignment*
 - c. Fungsi *Target A Goal*
 - d. Fungsi *Payroll*
 - e. Fungsi *Decruitment*
3. Sistem ERP yang dibangun dalam aspek *Multitenancy* dengan *distributed database*.
4. Platform menggunakan framework Yii2 dengan teknologi:
 - a. *Eloquent*
 - b. *Modularity*
 - c. *RBAC (Role Base Access Controll)*

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah:

1. Membuat proses bisnis modul Human Resource Management pada ERP.
2. Membuat aplikasi modul HRM.
3. Mengimplementasikan *multitenancy* pada modul HRM.
4. Mengatasi kelemahan *separated database* pada pengembangan sebelumnya dalam hal *availability* dan replikasi.

1.5 Manfaat

Manfaat dari hasil tugas akhir ini adalah memberikan kontribusi untuk membangun aplikasi ERP (Enterprise Resource Planning) yang berguna untuk kalangan UKM di Indonesia demi

meningkatkan pendapatan mereka. Dengan adanya aplikasi ini diharapkan aplikasi ERP lebih berkembang untuk penggunaan skala UKM.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

a. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi latar belakang pembuatan tugas akhir, rumusan masalah, batasan masalah, tujuan pembuatan, manfaat, metodologi hingga jadwal kegiatan pembuatan tugas akhir. Selain itu proposal tugas akhir ini memberikan ringkasan dari tugas akhir. Proposal tugas akhir juga berisi tinjauan pustaka yang digunakan sebagai referensi pembuatan tugas akhir ini.

b. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan program yaitu mengenai *Business Proccess Modelling and Notation*, *Yii2 Framework*, dan *ERP System*.

c. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

d. Implementasi perangkat lunak

Implementasi perangkat lunak ini dibangun dengan bahasa pemrograman PHP dan *database MySQL*. Selain itu untuk memudahkan pengembangan dan perbaikan di tahap selanjutnya, maka dibangun dengan salah satu kerangka kerja khusus PHP.

e. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan juga untuk mengevaluasi jalannya system, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode kotak hitam (*black box testing*) untuk mengetahui aspek nilai fungsionalitas dari perangkat lunak dan nilai kegunaan yang dibuat dengan juga memperhatikan ketertarikan pada calon partisipan untuk menggunakan aplikasi ini.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

BAB II DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1 Penelitian Terkait

Tugas Akhir ini merupakan pengembangan riset berkelanjutan dari riset atau penelitian tentang ERP. Pada Tugas Akhir sebelumnya telah dikembangkan sebuah ERP bernama ERP 2013. Pada ERP 2013 belum terdapat modul yang menangani sumber daya manusia. Maka dari itu pada pengembangan tahun 2016 ini ditambahkan sebuah modul yang menangani sumber daya manusia yakni HRM. Modul HRM menjalankan proses bisnis *Hired-To-Fired* dalam sistem ERP. Proses bisnis ini meliputi: *Recruitment and Hiring, Assignment, Training, Target A Goal, Payroll* dan *Decruitment* [1].

Tabel 2.1 Perbedaan Modul ERP 2013 dengan 2016

No.	Nama Modul ERP 2013	Nama Modul ERP 2016
1	Account Payable & Receivable	Account Payable & Receivable
2	General Ledger	General Ledger
3	Cost Accounting	Finance
4	Customer Relationship Management	Sales and Distribution
5	Production Planning	Production
6	Supplier Relationship Management	Purchasing
7	Inventory	Inventory and Warehouse Management
8	Assets Management	Assets Management
9	-	Human Resource Management

Pada tabel 2.1 dijabarkan modul-modul yang ada di pengembangan ERP 2013 dan ERP 2016. Pada ERP 2013 tidak ada modul HRM. Karena pada pengembangan sebelumnya, tidak ada sumber daya yang menangani khusus tentang HRM. Modul-modul yang digunakan pada Tugas Akhir ini mengacu pada ERP 2013 dengan cara menjalankan keseluruhan proses aplikasi ERP 2013.

Tabel 2.2 Sub Modul HRM

No	Nama Sub Modul HRM	Deskripsi
1	Recruitment and Hiring	Proses Perekrutan Pegawai Baru
2	Assignment	Proses administrasi Pegawai
3	Target A Goal	Proses Penentuan Bonus
4	Training	Proses Pengelolaan Pelatihan
5	Payroll	Proses Penggajian Pegawai
6	Decruitment	Proses Pengurangan Pegawai

Pada tabel 2.2 menjelaskan detail proses yang ada di dalam modul HRM. Proses-proses ini merupakan satu proses urut dari perekrutan sampai pengurangan pegawai [2].

2.2 ERP (*Enterprise Resource Planning*)

ERP adalah sebuah sistem diperuntukan bagi perusahaan manufaktur maupun jasa yang berperan mengintegrasikan dan mengotomasikan proses bisnis yang berhubungan dengan aspek operasi, produksi, maupun distribusi di perusahaan bersangkutan. ERP sendiri bermula pada kebutuhan bisnis sebuah perusahaan skala *enterprise* dalam menjalankan semua proses bisnisnya [2].

Keuntungan penggunaan ERP diantaranya adalah integrasi data keuangan, standarisasi proses operasi, standarisasi data dan informasi, penurunan inventori dan tenaga kerja, peningkatan servis dan kontrol keuangan dan penurunan waktu yang dibutuhkan untuk mendapatkan informasi. Ada pun departemen-departemen atau bagian-bagian yang pada umumnya terintegrasi meliputi:

- a. Pengelolaan keuangan (*financial management*) dalam aplikasi ERP 2016 ini diwakili oleh modul *Account Payable, Account Receivable, Fixed Asset, Cost Management, General Ledger*.
- b. Pengelolaan rantai pemasokan (*supply chain management*) diwakili oleh modul *Purchasing*.
- c. Perencanaan produksi dan manufaktur (*manufacturing resource planning*) diwakili oleh modul *Production dan Inventory*.
- d. Pengelolaan sumber daya manusia (*Human Resource Management*) diwakili oleh modul *Human Resource*.
- e. Pengelolaan relasi dengan pelanggan (*customer relationship management*) diwakili oleh modul *Sales and Distribution*.

2.3 Proses Bisnis *Human Resource Management*

Proses bisnis adalah kumpulan dari tugas-tugas atau aktivitas-aktivitas yang saling membentuk menjadi hasil yang diinginkan. Setiap proses dipicu oleh sebuah kejadian, seperti terjadinya pembelian oleh *customer* membuat penambahan barang di dalam gudang. Sistem enterprise merupakan sistem untuk membantu menyelesaikan proses-proses yang ada di dalam perusahaan [2].

Human Capital Management (HCM) terdiri dari banyak proses yang terkait dengan sumber daya manusia di semua aspek pengelolaan organisasi. Contoh proses HCM adalah *recruiting and hiring, training, compensation and benefits management*, dan *payroll administration* [2]. Kemudian sebagai dasar acuan detail proses di dalam modul HCM. Kami mengacu dalam aplikasi ERP (Odoo, Adempiere, dan InoERP).

2.3.1 *Recruitment and Hiring*

Proses mencari, mengidentifikasi dan menarik para calon atau kandidat pegawai yang berkemampuan. Proses ini membantu

perusahaan dalam menyeleksi atau mengambil pegawai baru yang sesuai dengan posisi yang dibutuhkan [3].

2.3.2 Assignment

Proses menempatkan dan menyimpan data pegawai yang ada di dalam perusahaan. Pada proses ini data pegawai berasal dari data kandidat yang telah diterima oleh perusahaan atau organisasi [3].

2.3.3 Training

Kegiatan manajemen Sumber Daya Manusia (SDM) untuk meningkatkan kemampuan dan keahlian pegawai [3].

2.3.4 Target A Goal

Proses penentuan jumlah bonus berdasarkan target yang telah dibuat oleh perusahaan atau organisasi. Proses ini mengacu pada proses kompensasi dan upah [3].

2.3.5 Payroll

Proses pemberian gaji untuk seluruh pegawai yang ada di dalam perusahaan.

2.3.6 Decruitment

Proses pengurangan pegawai berdasarkan beberapa jenis yang sudah ada seperti, memecat, merumahkan, membiarkan lowong, memindahkan, mengurangi jam, pensiun dini, dan berbagi pekerjaan [3].

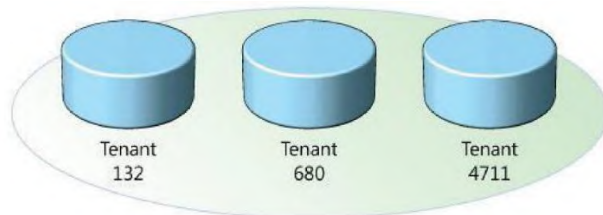
2.4 Multitenant Data Architecture atau Multitenancy

Sebuah arsitektur data pada sistem perangkat lunak dimana tenant memungkinkan mengakses data yang dimiliki sendiri dalam database yang sama. Arsitektur data ini didesain untuk *multitenancy* dimana setiap tenant mendapat *sharing* data, konfigurasi, manajemen pengguna, fungsionalitas setiap tenant, dan properti non-fungsional [4] [5].

Arsitektur data *multitenancy* digunakan karena kuat dan cukup aman untuk mengatasi masalah kurangnya kepercayaan *tenant* untuk menyerahkan kontrol data bisnis ke pihak ketiga, dan juga efektif dan efisien dalam pengelolaan dan pemeliharaan. Arsitektur data *multitenancy* terbagi menjadi tiga, yaitu:

2.4.1 *Separated Database, Separated Schema*

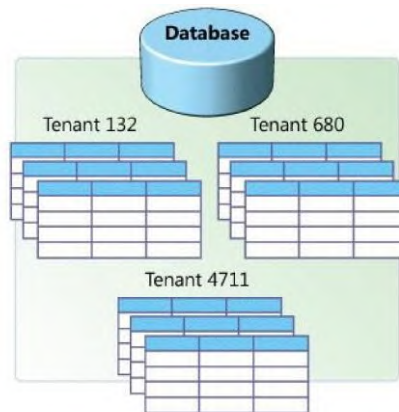
Pada *multitenant separated database*, setiap *tenant* mendapatkan masing-masing *database*, sehingga mudah untuk mengatur kembali model data aplikasi yang digunakan. Akan tetapi dalam penggunaan arsitektur ini memerlukan biaya yang tinggi untuk menjaga peralatan yang digunakan untuk server dan juga untuk back up data dari masing-masing *tenant* [3].



Gambar 2.1 *Separated Schema, Separated Database*

2.4.2 *Shared Database, Separated Schema*

Pada arsitektur *multitenant shared database*, *separate schema*, saat *tenant* menyewa ke servis, subsistem yang ada akan membuat tabel untuk *tenant* sesuai dengan skema yang dimiliki. Arsitektur ini mudah untuk digunakan (tabel yang dibentuk pada awalnya merupakan tabel standar, dan selanjutnya dapat diubah sesuai keinginan *tenant*), akan tetapi dalam apabila terjadi kegagalan, maka perlu dilakukan perbaikan untuk semua tabel yang ada di dalam *database* [6].



Gambar 2.2 *Shared Database, Separated Schema*

TenantID		CustName		Address	
4	TenantID	ProductID		ProductName	
1	4	TenantID	Shipment	Date	
6	1	4711	324965	2006-02-21	
4	6	132	115468	2006-04-08	
	4	680	654109	2006-03-27	
		4711	324956	2006-02-23	

Gambar 2.3 *Shared Schema, Shared Database*

2.4.3 *Shared Database, Shared Schema*

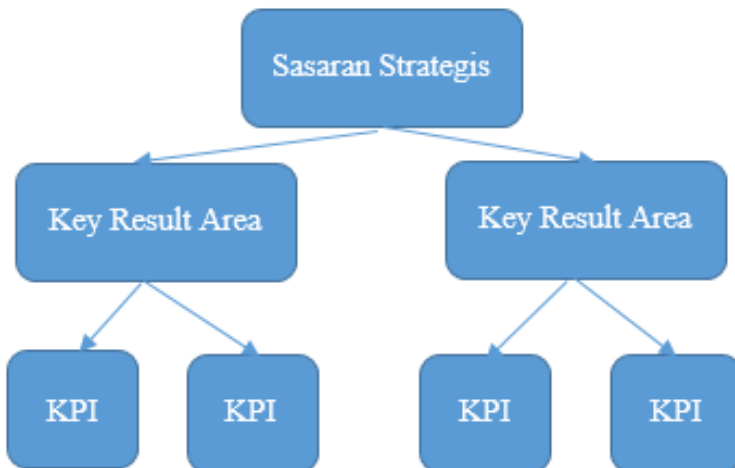
Pada arsitektur multitenant *shared database, shared schema*, menggunakan *database* dan tabel yang sama untuk banyak tenant, untuk membedakan masing-masing tenant digunakan *id tenant* untuk setiap tabel yang ada. Penggunaan model ini tidak memerlukan biaya yang tinggi, akan tetapi apabila terjadi kegagalan, akan sama seperti dengan model multitenant *shared database, separate schema* [6].

2.5 *Human Resource Management (HRM)*

HRM merupakan salah satu modul yang ada di dalam aplikasi ERP. Modul ini menangani pengelolaan SDM untuk mencapai bisnis gol perusahaan atau organisasi. Pengelolaan SDM mencakup fungsi-fungsi utama: *Recruitment and Hiring*, *Assignment*, *Training*, *Target A Goal*, *Payroll* dan *Decruitment*. Di dalam proses HRM memiliki bermacam-macam aktivitas yang berbeda tergantung pada perusahaan. Namun secara garis besar ada pada ke enam sub modul di atas [2] [7].

2.6 **KPI (Key Performance Indicator)**

KPI adalah sebuah matrik finansial atau non finansial yang digunakan oleh perusahaan untuk mengukur performa kinerja mereka. Di dalam KPI terdapat bobot, target, realisasi, skor dan final skor. Skema KPI dapat dilihat dalam bentuk gambar berikut:



Gambar 2.4 Skema *Key Performance Indicator*

2.7 **Kriteria SMART**

SMART merupakan singkatan dari kata *Specific*, *Measurable*, *Achieveable*, *Relevant* dan *Time-bound*. *Specific*

berarti sasaran kerja harus bersifat rinci dan detail. *Measurable* berarti terukur atau dapat diukur dengan satuan yang jelas. Konsek *Measurable* ini juga digunakan penentuan *Key Performance Indicator*. *Achievable* artinya target yang ditetapkan masih realistis atau sesuai dengan sumber daya yang tersedia. *Relevant* berarti sasaran atau tujuan yang ingin dicapai bersifat relevan antara tugas pokok dan tanggungjawab yang diemban oleh pegawai. Dan yang terakhir adalah *Time-bound* yang berarti sasaran yang telah disusun memiliki target waktu yang sangat jelas dan disiplin [8].

2.7.1 *Specific*

Target memiliki sifat yang spesifik, detail, dan jelas. Hal ini diperlukan agar target Key Performance Indicator yang dibuat tidak menimbulkan abiguitas.

2.7.2 *Measurable*

Terukur, sebuah kriteria dimana sebuah target atau KPI yang dibuat harus dapat diukur. Hal ini membantu sebuah bisnis goal yang abstrak dapat diukur dengan sebuah angka.

2.7.3 *Achievable*

Target yang telah ditetapkan merupakan hal yang realistis dan dapat dicapai dan ada nilai atau hasil peningkatan.

2.7.4 *Relevant*

Target yang dipilih merupakan aspek-aspek yang relevan dan berkaitan dengan tugas pokok. Hal ini bertujuan agar penetapan target sesuai dengan apa yang diinginkan perusahaan.

2.7.5 *Time-bound*

Waktu untuk mencapai target tersebut. Waktu itu dapat berupa berapa lama target dapat tercapai dan kapan target itu dicapai. Waktu ini dapat juga menjadi tolok ukur yang baik.

2.8 Pajak Penghasilan Pph 21

Sesuai dengan Pasal 17 ayat 1, Undang-Undang No. 36 tahun 2008, tarif pajak penghasilan pribadi perhitungannya dengan menggunakan tarif progresif sebagai berikut [7]:

Tabel 2.3 Penghasilan Netto Kena Pajak

Penghasilan Netto Kena Pajak	Tarif Pajak
Sampai dengan 50 Juta	5%
50 juta sampai dengan 250 juta	15%
250 juta sampai dengan 500 juta	25%
Diatas 500 juta	30%

PTKP 2016 bagi Wajib Pajak Tidak Kawin dan memiliki tanggungan:

Tabel 2.4. PTKP Kelompok Pertama

Uraian	Status	PTKP
Wajib Pajak	TK0	54.000.000,-
+ Tanggungan 1	TK1	58.500.000,-
+ Tanggungan 2	TK2	63.000.000,-
+ Tanggungan 3	TK3	67.500.000,-

PTKP 2016 bagi Wajib Pajak Kawin dan memiliki anak / tanggungan.

Tabel 2.5. PTKP Kelompok Kedua

Uraian	Status	PTKP
+ WP Kawin	K0	58.500.000,-
+ Kawin Anak 1	K1	63.000.000,-
+ Kawin Anak 2	K2	67.500.000,-
+ Kawin Anak 3	K3	72.000.000,-

Catatan:

- Tunjangan PTKP untuk anak atau tanggungan maksimal 3 orang
- TK : Tidak Kawin.

- K : Kawin.
- K/I : Kawin dan penghasilan pasangan digabung.

2.9 *Distributed Database*

Basis data di mana perangkat penyimpanan tidak semua terpasang ke unit pengolahan umum seperti CPU (*Central Proccessing Unit*). Ini dapat disimpan dalam beberapa komputer yang terletak di lokasi fisik yang sama, atau dapat tersebar melalui jaringan komputer yang saling berhubungan. Tidak seperti sistem paralel, di mana prosesor yang erat dan merupakan sebuah sistem database tunggal, sistem basis data terdistribusi terdiri dari situs longgar ditambah yang berbagi tidak ada komponen fisik [4].

2.9.1 Replikasi

Replikasi adalah mekanisme penyalinan seluruh atau sebagian tabel basis data ke beberapa *server data*. Setiap transaksi (penambahan, penghapusan, atau perubahan data) akan dieksekusi pada semua *server data* yang menyusun sistem BDT. Hal ini menyebabkan mekanisme replikasi membebani performa sistem, namun memiliki tingkat kompleksitas yang paling sederhana.

2.9.2 Fragmentasi

Fragmentasi adalah mekanisme penyalinan sebagian data atau struktur dari setiap tabel basis data. Transaksi basis data yang terjadi harus diolah terlebih dahulu oleh *server manajemen* untuk menentukan letak *server data* dari tabel yang akan terpengaruh. Hal ini menyebabkan mekanisme fragmentasi memiliki tingkat kompleksitas yang tinggi, namun memiliki performa yang lebih baik dibandingkan dengan mekanisme replikasi.

2.10 *MySQL Cluster*

MySQL Cluster merupakan sebuah tipe basis data yang dapat beroperasi dalam ukuran data yang relatif besar. *MySQL Cluster* adalah sebuah teknologi baru untuk memungkinkan *clustering* di dalam *memory* basis data dalam sebuah sistem *share-nothing*. Arsitektur *share-nothing* mengijinkan sistem dapat

bekerja dengan perangkat keras yang sangat murah, dan tidak membutuhkan perangkat keras dan lunak dengan spesifikasi khusus. Arsitektur tersebut juga handal karena masing-masing komponen mempunyai *memory* dan *disk* tersendiri. *MySQL Cluster* menggabungkan *MySQL Server* biasa dengan sebuah mesin penyimpanan *in-memory* ter-*cluster* yang dinamakan NDB. NDB berarti bagian dari suatu rangkaian yang dikhususkan sebagai mesin penyimpanan, sedangkan *MySQL Cluster* diartikan sebagai kombinasi atau gabungan dari *MySQL* dan mesin penyimpanan yang baru tersebut¹.

2.11 PHP (*Hypertext Preprocessor*)

PHP adalah singkatan dari *Hypertext Preprocessor* yakni salah satu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari Personal Home Page Tools. Selanjutnya diganti FI (Forms Interpreter). Sejak versi 3.0, nama bahasa ini diubah menjadi PHP (*Hypertext Preprocessor*) dengan singkatan PHP. PHP versi terbaru adalah versi ke-7².

2.12 Kerangka Kerja Yii2

Sebuah *framework* untuk bahasa pemrograman PHP dimana *framework* ini telah didukung oleh aspek *modularity* dan *eloquent database*. Arsitektur perangkat lunak ini menggunakan *Model-View-Controller*, namun dengan aspek *modularity*, maka dapat dibuat modul dimana setiap modul memiliki MVC tersendiri. Sedangkan *eloquent database* adalah basisdata berupa objek di dalam sebuah bahasa pemrograman PHP³.

¹ <https://www.mysql.com/products/cluster/>

² <https://id.wikipedia.org/wiki/PHP>

³ <http://www.yiiframework.com/doc-2.0/guide-intro-yii.html>

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Bab ini membahas tahap analisis dan perancangan sistem yang akan dibangun. Analisis membahas semua persiapan yang akan menjadi pokok pikiran pembuatan aplikasi ini. Mulai dari masalah yang melatarbelakangi, hingga analisis gambaran awal sistem yang akan dibuat. Perancangan sistem membahas hal-hal yang berkaitan dengan pondasi atau dasar pembuatan aplikasi, yang meliputi perancangan basis data, tampilan antar muka halaman aplikasi, hingga perancangan alur proses yang akan diimplementasikan di dalam aplikasi.

3.1 Analisis

Tahap analisis dibagi menjadi beberapa bagian antara lain: domain permasalahan, deskripsi umum sistem, dan kasus penggunaan sistem. Berikut pembahasan bagian-bagian tahap analisis.

3.1.1 Proses Bisnis

Aplikasi ERP bukan aplikasi yang baru muncul atau jarang digunakan. Namun aplikasi ERP merupakan aplikasi yang sudah lama ada. Namun masing-masing ERP memiliki proses bisnis yang berbeda-beda. Kebanyakan aplikasi ERP yang kompleks dan sesuai dengan banyak perusahaan memiliki harga jual yang tinggi. Permasalahan berada pada bagaimana business plan yang telah ada dapat dijalankan secara efisien dengan aplikasi ERP ini. Setiap ERP memiliki proses bisnis yang berbeda-beda. Aplikasi ERP pada Tugas Akhir ini memiliki proses bisnis sendiri dan dibandingkan dengan aplikasi ERP yang cukup banyak digunakan banyak perusahaan. Seperti Odoo, Adempiere, dan InoERP. Pada bab ini akan dijelaskan tentang analisa proses bisnis yang telah ada.

Tabel 3.1 Kekurangan dan Kelebihan pada Odoo, Adempiere, dan InoERP

Nama Modul	Odoo	Adempiere	InoERP
Sales and Delivery	Kontra: Pada proses creating delivery, apabila barang yang dipesan tidak tersedia di gudang/storage, delivery tetap dapat dilakukan	Contra: 1. Pada Proses Creating delivery, apabila barang yang dipesan tidak tersedia di gudang/storage, delivery tetap dapat dilakukan	Contra: 1. Pada Proses Creating delivery, apabila barang yang dipesan tidak tersedia di gudang/storage, delivery tetap dapat dilakukan
Inventory	-	kontra: 1. Desain inventory pada InoERP: Warehouse->Locator->Storage Desain inventory pada EZERP: Warehouse->Inventory->Storage->SubStorage->Bin	kontra: 1. Desain inventory pada InoERP: Inventory->SubInventory->Locator Desain inventory pada EZERP: Warehouse->Inventory->Storage->SubStorage->Bin

Nama Modul	Odoo	Adempiere	InoERP
			2. Pencatatan pengiriman barang diluar gudang tidak ada
Purchasing	Kontra: tidak ada sub modul purchase requisition, tidak ada analisa terhadap transaksi yang telah dilakukan sebelumnya	Kontra: tidak ada modul untuk mengurus quotation dari supplier, tidak ada analisa terhadap transaksi yang telah dilakukan sebelumnya	Pro: Fitur purchasing yang terdapat di INOERP sudah lengkap Kontra: Tidak ada analisa terhadap transaksi yang sebelumnya telah dilakukan
Account Receivable and Account Payable	Kontra: tidak dapat melakukan pembayaran secara kredit Kontra: tidak dapat melakukan pembayaran secara kredit	-	Pro: Fitur sudah lengkap Kontra: tidak ada analisa transaksi yang pernah dilakukan perusahaan Pro: Fitur sudah lengkap Kontra: tidak ada analisa

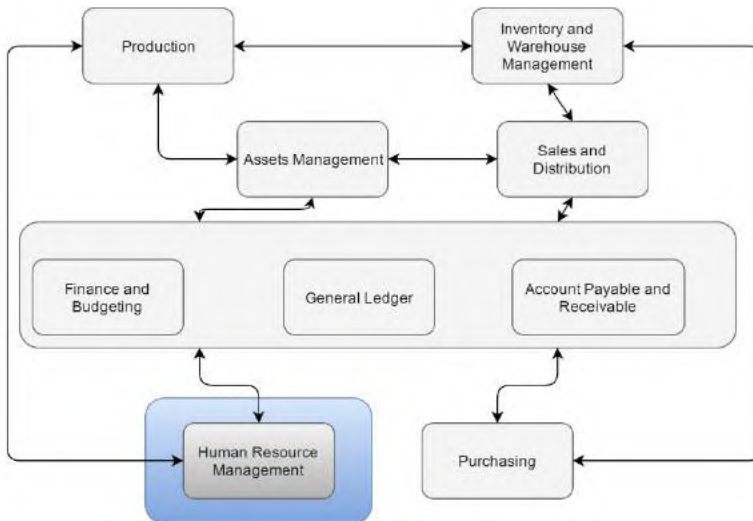
Nama Modul	Odoo	Adempiere	InoERP
			transaksi yang pernah dilakukan customer
Production	Kontra : Tidak ada routing, sehingga penggunaan sumberdaya (mesin dan manusia) tidak tercatat menjadi cost.	Kontra: Scheduling masih manual (tidak otomatis)	Kontra: Scheduling masih manual (tidak otomatis)
Human Resource Management	Kontra : Tidak terdapat KPI dan Payroll masih manual , tidak otomatis.	Kontra : Tidak terdapat KPI dan Payroll masih manual , tidak otomatis.	Kontra : Tidak terdapat KPI dan Payroll masih manual , tidak otomatis.
Assets Management	Cons : 1. Produksi tidak melakukan pengecekan status availability mesin. 2. Maintenance tidak berdasarkan	Kontra : Maintenance tidak mengikuti waktu produksi	Pro : Fitur maintenance yang terdapat pada INOERP sudah lengkap terdapat BOM untuk setiap mesin, jadi saat mesin maintenance hanya perlu mengganti

Nama Modul	Odoo	Adempiere	InoERP
	waktu produksi		partisi yang dibutuhkan Cons: Maintenance tidak berbasis pada waktu produksi, sistem tidak memberikan warning apabila mesin butuh di maintenance
Finance	Kontra : Tidak terdapat COGM maupun COGS sehingga pada modul SD manual	Pro : COGM sudah detail Kontra : Expense atau overhead tidak ada	Pro : Fitur costing yang terdapat pada INOERP sudah lengkap Kontra: Masih manual dalam proses pemberian harga barang jadi, tidak melalui proses COGM, COGS dan overhead

Tabel 3.1 menunjukkan bahwa masih banyak kekurangan atau tidak memenuhi kebutuhan dari *business plan* yang telah dibuat. Pada modul HRM dapat dilihat bahwa modul HR pada Odoo, Adempiere dan InoERP belum ada submodul KPI dalam

menentukan bonus. Selain itu untuk penggajian pegawai masih secara manual satu per satu.

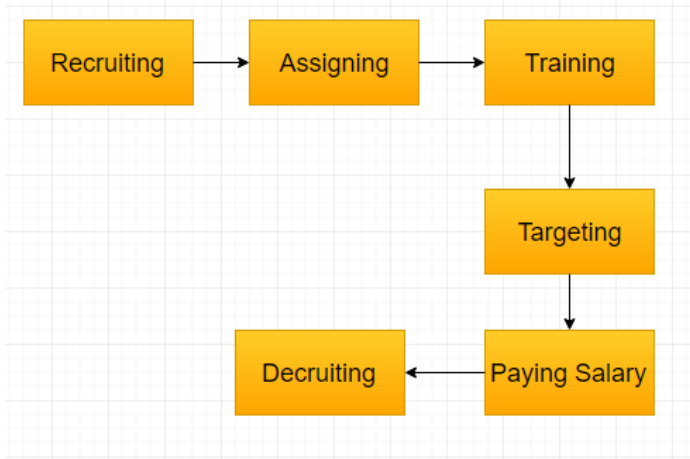
Berdasarkan kekurangan yang dimiliki oleh setiap ERP maka perlu dikembangkannya sebuah aplikasi ERP yang sesuai dengan kebutuhan pada *business plan*. Terdapat dua jenis proses bisnis yang digunakan, yakni *Make-To-Order* (MTO) dan *Make-To-Stock* (MTS). MTO adalah sebuah proses bisnis dimana produksi didasarkan pada permintaan. MTS adalah sebuah proses bisnis dimana produksi didasarkan oleh *sales-forecasting*. Berikut proses bisnis yang dirancang sesuai dengan kebutuhan *business plan* dimana proses bisnis ini telah mengakomodasi kekurangan Odoo, Adempiere, dan InoERP.



Gambar 3.1 Workflow ERP Level 0

Pada Gambar 3.1 digambarkan hubungan antar modul ERP. Level 0 ini menggambarkan hubungan antar modul dan modul apa yang difokuskan di dalam Tugas Akhir ini, yakni modul *Human Resource Management*. Modul ini menangani masalah pengelolaan

sumber daya manusia di dalam perusahaan atau organisasi dimana memiliki fungsi-fungsi yang telah dijelaskan pada bab 2.2.



Gambar 3.2 Workflow ERP Level 1 pada modul HRM

Pada Level 1 digambarkan sub-modul yang ada di dalam modul-modul ERP.

Tabel 3.2 Nama Modul Level 0

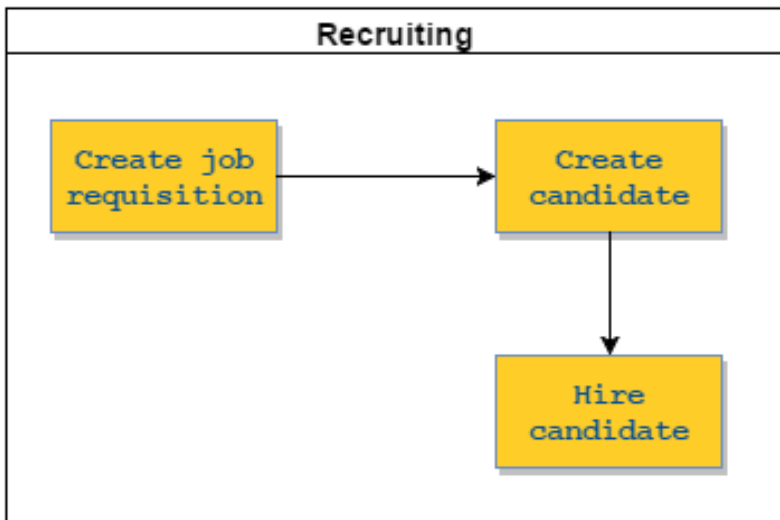
Kode	Nama Modul
1	Assets Management
2	Production
3	Warehouse Management
4	Purchasing
5	Accounting
6	Human Resource
7	Sales and Distribution

Pada Tabel 3.2 kode nomer 5 dikelompokkan dengan nama *Accounting*. Di dalam *Accounting* terdapat 3 modul utama yakni, *Account Payable and Account Receivable*, *General Ledger* dan *Finance*.

Tabel 3.3 Nama Submodul Level 1

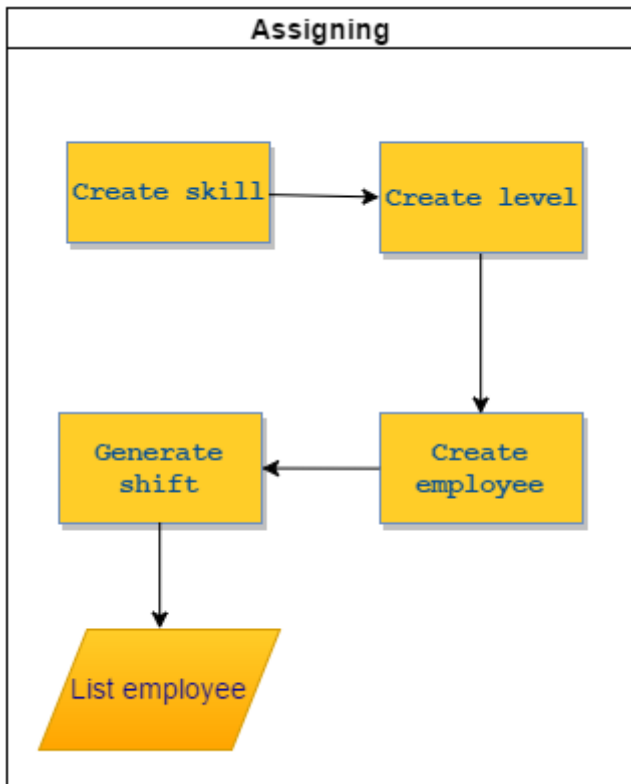
Kode	Nama Submodul HRM
6.1	Recruitment and Hiring
6.2	Assignment
6.3	Training
6.4	Target A Goal
6.5	Payroll
6.6	Decruitment

Pada Level 2 dijelaskan bagaimana proses yang terjadi dari setiap submodul yang ada di Tabel 3.3. Berikut gambar level 2 dari setiap submodul:

**Gambar 3.3 Recruiting**

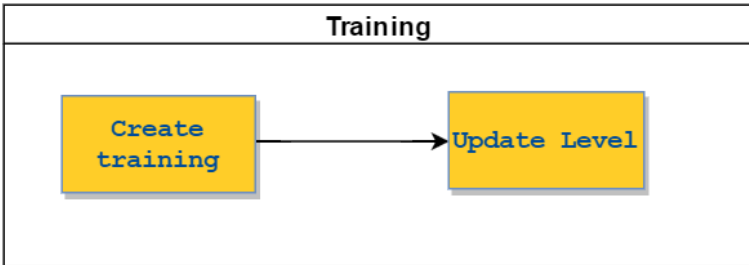
Pada Gambar 3.3 menggambarkan proses perekrutan pegawai secara sistem. Perekrutan pegawai dimulai dari pembuatan Job Requisition dimana hal ini berdasarkan posisi yang

sedang kosong atau dibutuhkan. Setelah itu dilanjutkan dengan pembuatan kandidat atau data calon pelamar posisi yang sedang kosong.



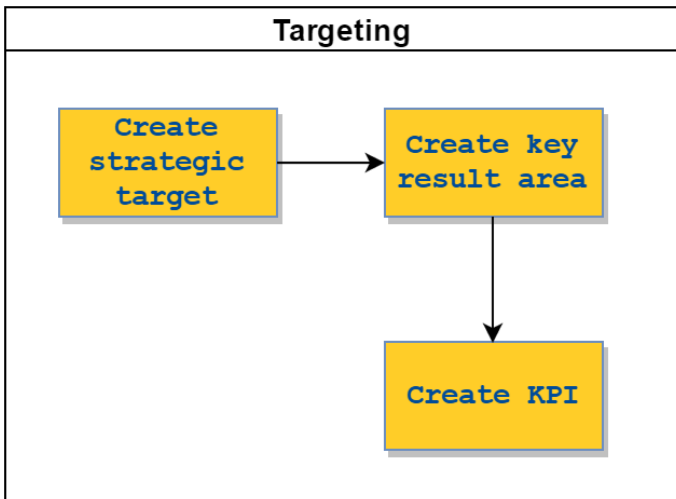
Gambar 3.4 Assigning

Pada Gambar 3.4 menggambarkan proses administrasi pegawai di dalam perusahaan. Proses ini merupakan lanjutan dari proses perekrutan pegawai. Data yang digunakan merupakan data yang berasal dari data kandidat yang telah diterima.



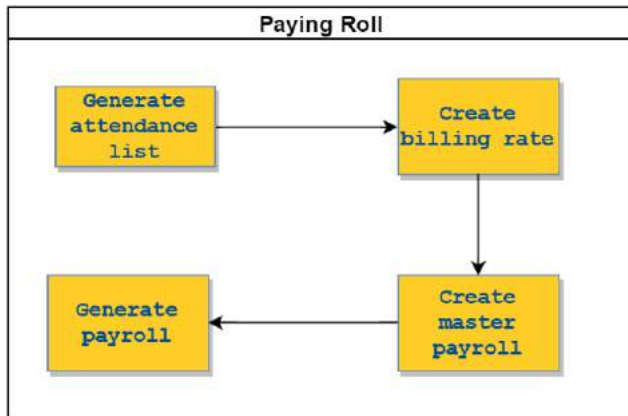
Gambar 3.5 Training

Pada Gambar 3.5 merupakan proses pelatihan pegawai. Pelatihan ini digunakan untuk meningkatkan keahlian pegawai.

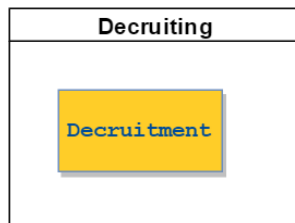


Gambar 3.6 Targeting

Pada Gambar 3.6 merupakan proses pembuatan goal perusahaan khususnya di bidang sumber daya manusia. Proses awal ialah pembuatan target sasaran yang bersifat luas. Proses selanjutnya adalah pembagian dalam bidang-bidang dalam *key result area*. Proses terakhir ialah membuat *Key Performance Indicator* yang telah dijelaskan pada bab 2.7.



Gambar 3.7 Paying Roll



Gambar 3.8 Decruiting

Pada Gambar 3.7 merupakan gambar proses penggajian. Proses ini dimulai dari input data kehadiran, kemudian *billing rate*, *master payroll* dan *generate payroll*. Dan pada Gambar 3.8 merupakan proses pengurangan pegawai. Untuk detail alur telah dilampirkan pada Lampiran A.1.

3.1.2 Analisa Data

Pada sub bab 2.2 telah dijelaskan pengertian proses bisnis berdasarkan buku “Integrated Business Processes with ERP Systems” bahwa proses bisnis merupakan kumpulan dari aktivitas-aktivitas atau tugas-tugas yang menjadi satu yang mengeluarkan sebuah hasil yang diinginkan. Dari dasar teori tersebut dapat dibuat sebuah proses bisnis dari sebuah sistem. Karena tingkat

kompleksitas yang tinggi pada proses bisnis sistem enterprise. Maka pada tugas akhir ini dibuat pengelompokan proses bisnis mulai dari level 0 sampai level 2.

Pada Tugas Akhir ini menggunakan sistem basis data dimana untuk setiap tenant memiliki Database dan Schema (tabel) yang berbeda. Pada dasarnya penggunaan basis data bisa memilih 3 jenis arsitektur basis data yang telah dijelaskan di bab 2.4.1 Penggunaan *Separated Database*, *Separated Schema* bertujuan untuk menunjang aspek *multitenancy* dan penggunaan basis data terdistribusi. Selain itu pemilihan *Separated Database*, *Separated Schema* memiliki keuntungan yang tidak dimiliki oleh yang lain, yakni pengaturan jauh lebih mudah, karena tidak menggabungkan semua tabel tenant pada satu basis data dan jika satu tenant mati, tenant yang lain masih dapat dijalankan.

Pada Tugas Akhir ini dikembangkan sebuah sistem basis data terdistribusi dimana menggunakan aspek *distributed* dan sinkronisasi basis data. Sistem distribusi merupakan sistem yang membagi data ke beberapa resource yang ada. Dalam tugas akhir ini membagi basis data pada tenant ke dalam beberapa node penyimpanan basis data. Sedangkan proses sinkronisasi merupakan fitur yang digunakan untuk menangani masalah ketika satu node mati, dan kemudian dihidupkan kembali. Maka node yang sebelumnya mati dapat mensinkronkan data-data yang telah diupdate pada node-node yang lain. Pada Tugas Akhir ini menggunakan 2 buah node penyimpanan.

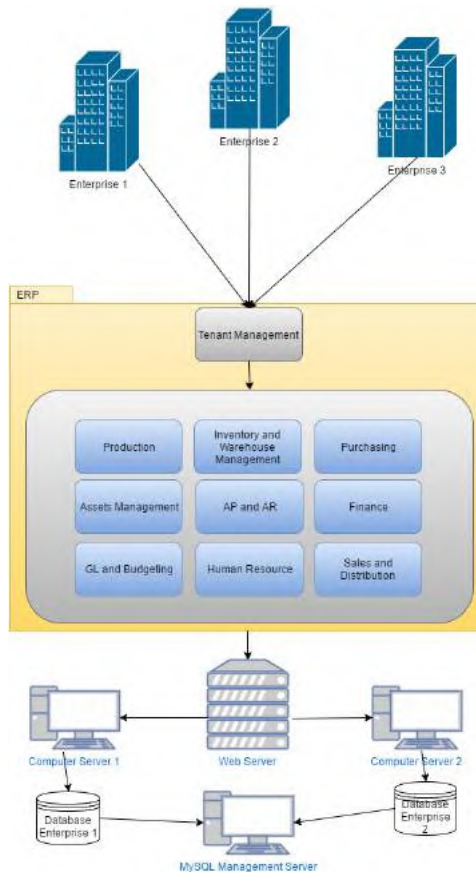
3.2 Deskripsi Umum Sistem

Pada Tugas Akhir ini dibangun aplikasi *Human Resource Management* yang terintegrasi ke dalam sebuah sistem Enterprise Resource Planning (ERP). Tujuan dari aplikasi *Human Resource Management* adalah untuk mengelola dan melakukan monitoring terhadap sumberdaya manusia dalam mencapai bisnis goal perusahaan. Sistem yang dibangun berorientasi *multi-tenancy* dengan basis data terdistribusi.

3.3 Identifikasi Pengguna

Pada sistem yang akan dibangun, aktor yang akan menjadi pengguna sistem adalah *staff* HRM dan *manager* HRM. Staff HRM mempunyai hak ases dalam pengelolaan sumber daya manusia, yang terdiri atas membuat (*create*), memperbarui (*update*) dan menghapus(*delete*). Sedangkan manager mempunyai hak akses melihat (*view*) data yang telah dikelola oleh staff.

3.4 Arsitektur Sistem



Gambar 3.9 Arsitektur Sistem

Pada Gambar 3.9, dijelaskan bahwa perusahaan 1, perusahaan 2, dan perusahaan 3 dapat menggunakan ERP secara bersamaan. Setiap perusahaan mempunyai 2 basis data yang telah direplikasi yaitu *Computer Server 1* dengan *IP Enterprise 1*, dan *Computer Server 2* dengan *IP Enterprise 2*. komputer 3 berfungsi sebagai *server cluster* untuk melakukan replikasi, sehingga apabila basis data 1 atau *Computer Server 1* dalam keadaan mati, sistem ERP masih tetap berjalan.

3.5 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan dalam sistem ini mencakup kebutuhan fungsional. Kebutuhan fungsional berisikan proses-proses bisnis yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.4

3.6 Kasus Penggunaan

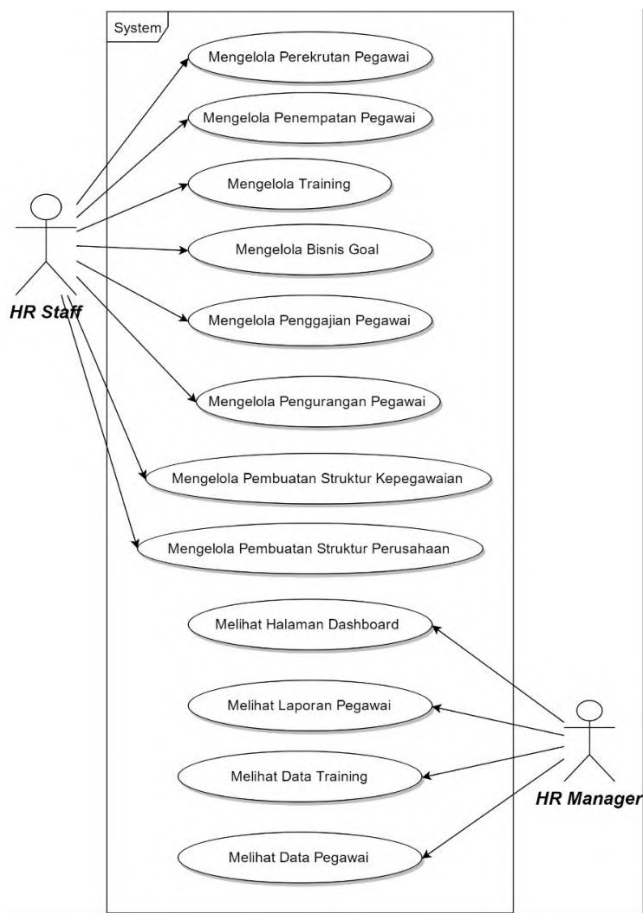
Kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.3 dan kode kasus penggunaan ada pada tabel 3.7.

Tabel 3.4 Daftar Kebutuhan Fungsional Sistem

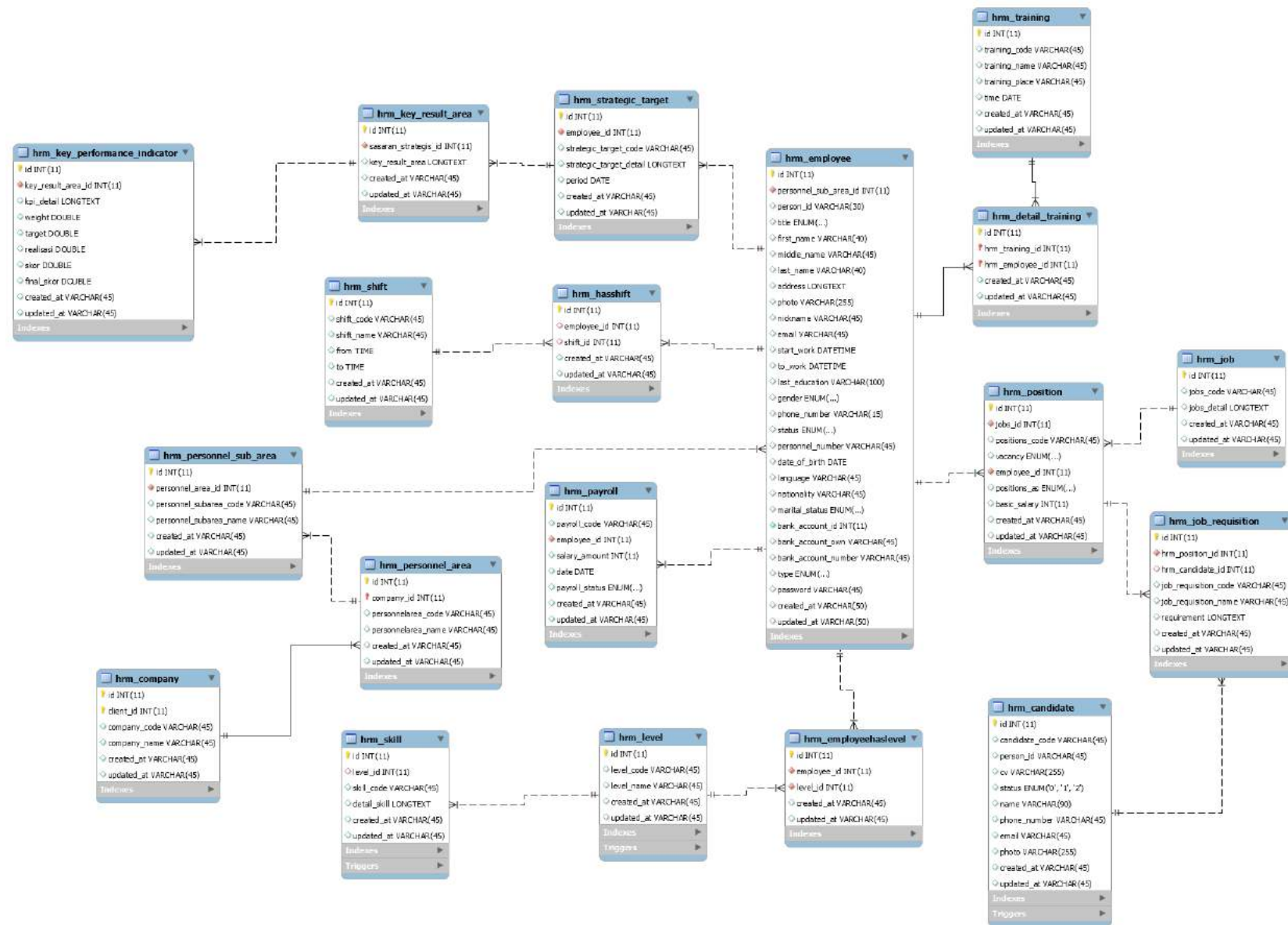
Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
FHRM-001	Mengelola Perekrutan Pegawai	Pengguna dapat mengelola perekrutan pegawai yang meliputi proses: <i>create, update, delete, view</i>
FHRM-002	Mengelola Penempatan Pegawai	Pengguna dapat mengelola penempatan pegawai yang meliputi proses: <i>create, update, delete, view</i>

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
FHRM-003	Mengelola Training	Pengguna dapat mengelola training yang meliputi proses: <i>create, update, delete, view</i>
FHRM-004	Mengelola Bisnis Goal	Pengguna dapat mengelola bisnis goal yang meliputi proses: <i>create, update, delete, view</i>
FHRM-005	Mengelola Penggajian Pegawai	Pengguna dapat mengelola penggajian yang meliputi proses: <i>create, update, view</i>
FHRM-006	Mengelola Pengurangan Pegawai	Pengguna dapat mengelola pengurangan pegawai yang meliputi proses: <i>create, update, delete</i>
FHRM-007	Mengelola Pembuatan Struktur Kepegawaian	Pengguna dapat mengelola pembuatan struktur pegawai yang meliputi proses: <i>create, update, delete, view</i>
FHRM-008	Mengelola Pembuatan Struktur Perusahaan	Pengguna dapat mengelola pembuatan struktur perusahaan yang meliputi proses: <i>create, update, delete, view</i>
FHRM-009	Melihat Halaman Dashboard	Pengguna dapat melihat halaman <i>dashboard</i>
FHRM-010	Melihat Laporan Pegawai	Pengguna dapat melihat halaman laporan pegawai

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
FHRM-011	Melihat Data Training	Pengguna dapat melihat data <i>training</i>
FHRM-012	Melihat Data Pegawai	Pengguna dapat melihat data pegawai.



Gambar 3.10 Diagram Kasus Penggunaan



Gambar 3.11 Physical Data Model

[Halaman ini sengaja dikosongkan]

Tabel 3.5 Keterangan Kode Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan
UCTA-001	Mengelola Pembuatan Struktur Perusahaan
UCTA-002	Mengelola Pembuatan Struktur Kepegawaian
UCTA-003	Mengelola Perekrutan Pegawai
UCTA-004	Mengelola Penempatan Pegawai
UCTA-005	Mengelola <i>Training</i>
UCTA-006	Mengelola Bisnis Goal
UCTA-007	Mengelola Penggajian Pegawai
UCTA-008	Mengelola Pengurangan Pegawai
UCTA-009	Melihat Halaman Dashboard
UCTA-010	Melihat Laporan Pegawai
UCTA-011	Melihat Data Training
UCTA-012	Melihat Data Pegawai

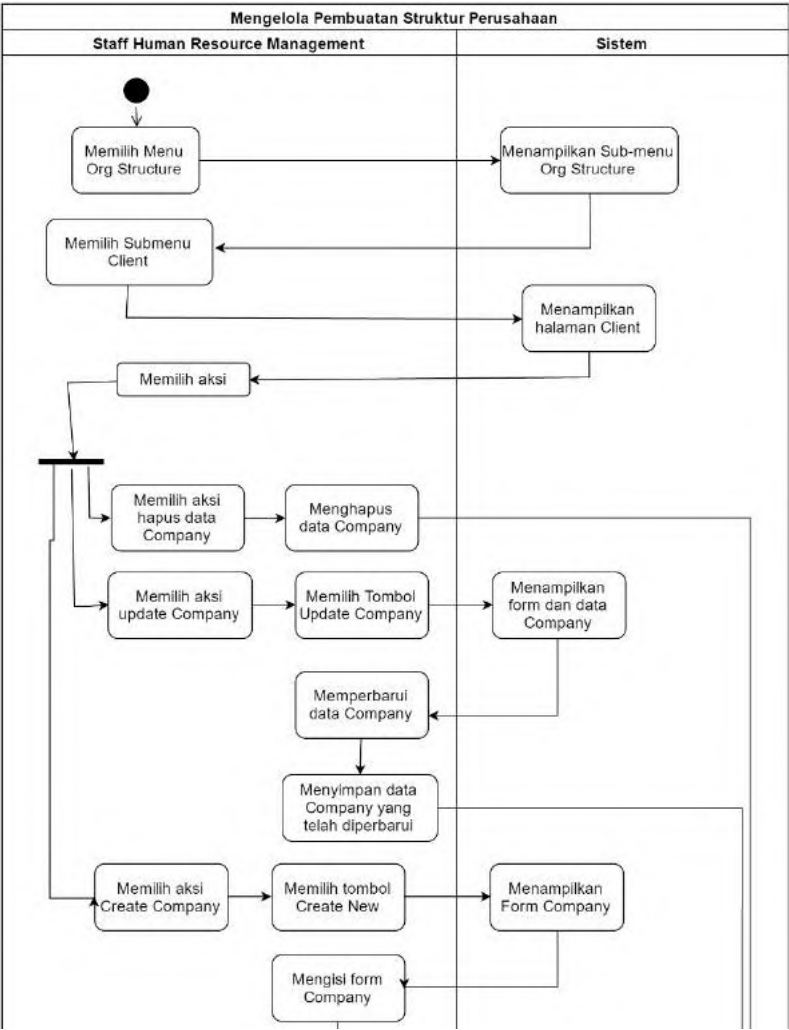
3.6.1 Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan

Pada kasus penggunaan ini, sistem menampilkan menu untuk melakukan pembuatan struktur perusahaan antara lain: *Company*, *Personnel Area*, dan *Personnel Sub Area*. Data ini merupakan data master yang digunakan untuk kasus penggunaan yang lain.

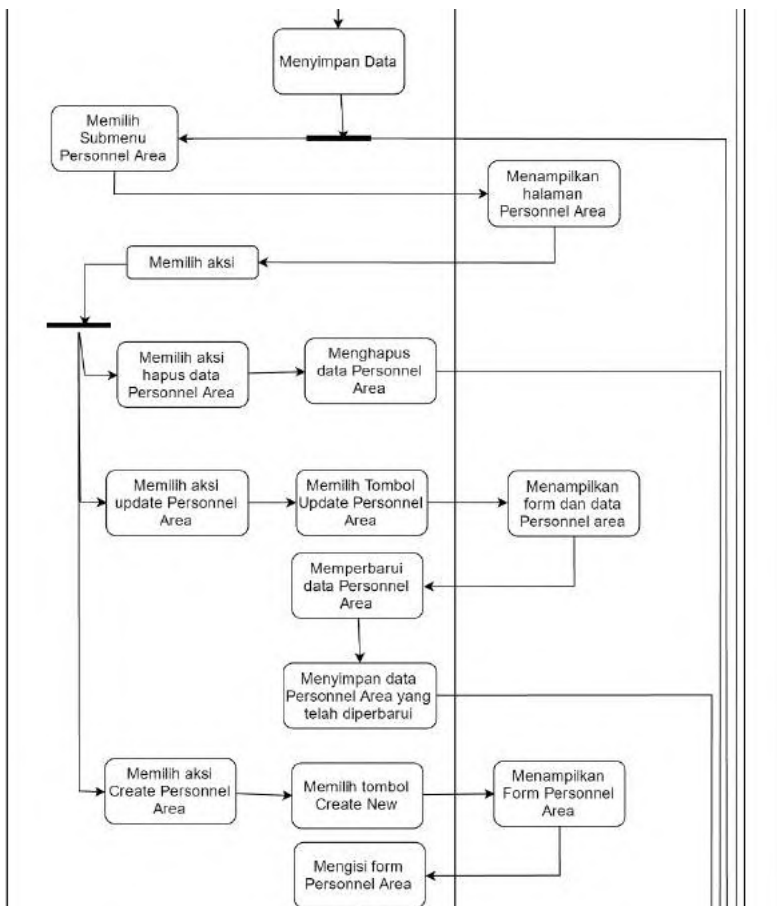
Tabel 3.6 Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan

Nama	Mengelola Pembuatan Struktur Perusahaan
Kode	UCTA-001
Deskripsi	Pengguna memilih menu <i>Org Structure</i> . Menu ini digunakan untuk membuat struktur perusahaan mulai dari <i>Company</i> , <i>Personnel Area</i> dan <i>Personnel Sub Area</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan tombol “Create”
Aktor	Staff <i>Human Resource Management</i>
Kondisi Awal	-
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan menu <i>Org Structure</i>. 2. Pengguna memilih menu <i>Company</i>. 3. Sistem menampilkan halaman <i>Company</i>. 4. Pengguna memilih aksi <i>Create</i>. 5. Sistem menampilkan form <i>Company</i>. 6. Pengguna mengisi form <i>Company</i>. 7. Pengguna menyimpan data <i>Company</i>. 8. Sistem menyimpan data ke dalam basis data. 9. Pengguna memilih menu <i>Personnel Area</i>. 10. Sistem menampilkan halaman <i>Personnel Area</i>. 11. Pengguna mengisi form <i>Personnel Area</i>. 12. Pengguna menyimpan data <i>Personnel Area</i>.

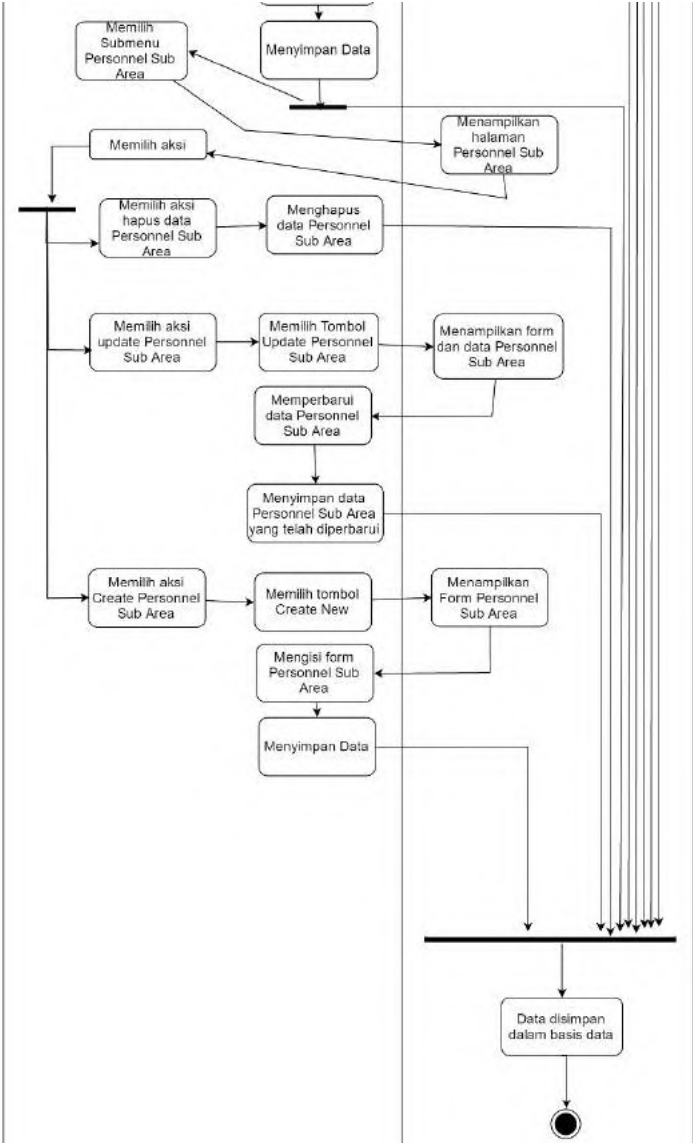
	<p>13.Sistem menyimpan data <i>Personnel Area</i> ke dalam basis data.</p> <p>14.Pengguna memilih menu <i>Personnel Sub Area</i>.</p> <p>15.Sistem menampilkan halaman <i>Personnel Sub Area</i>.</p> <p>16.Pengguna memilih aksi <i>Create</i>.</p> <p>17.Sistem menampilkan form <i>Personnel Sub Area</i>.</p> <p>18.Pengguna mengisi form <i>Personnel Sub Area</i>.</p> <p>19.Pengguna menyimpan data <i>Personnel Sub Area</i>.</p> <p>20.Sistem menyimpan data <i>Personnel Sub Area</i> ke dalam basis data.</p>
- Kejadian Alternatif	<p>2a. Pengguna memilih menu <i>Personnel Area</i></p> <p>2b. Pengguna memilih menu <i>Personnel Sub Area</i>.</p> <p>Sistem kembali ke kejadian normal 2.</p> <p>9a. Pengguna memilih menu <i>Company</i></p> <p>9b. Pengguna memilih menu <i>Personnel Sub Area</i>.</p> <p>Sistem kembali ke kejadian normal 9.</p> <p>16a. Pengguna memilih menu <i>Company</i></p> <p>16b. Pengguna memilih menu <i>Company</i>.</p> <p>Sistem kembali ke kejadian normal 16</p>
Kondisi Akhir	Data struktur perusahaan masuk ke dalam basis data.



Gambar 3.12 Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan Bagian (1)



Gambar 3.13 Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan Bagian (2)



Gambar 3.14 Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Perusahaan Bagian (3)

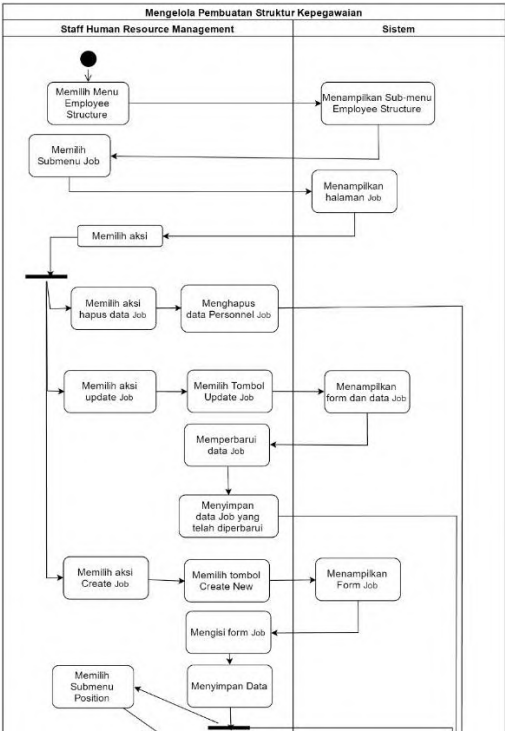
3.6.2 Kasus Penggunaan Mengelola Pembuatan Struktur Kepegawaian

Pada kasus penggunaan ini, sistem menampilkan menu untuk melakukan pembuatan struktur kepegawaian antara lain: *Job*, dan *Position*. Data ini merupakan data master yang digunakan untuk kasus penggunaan yang lain..

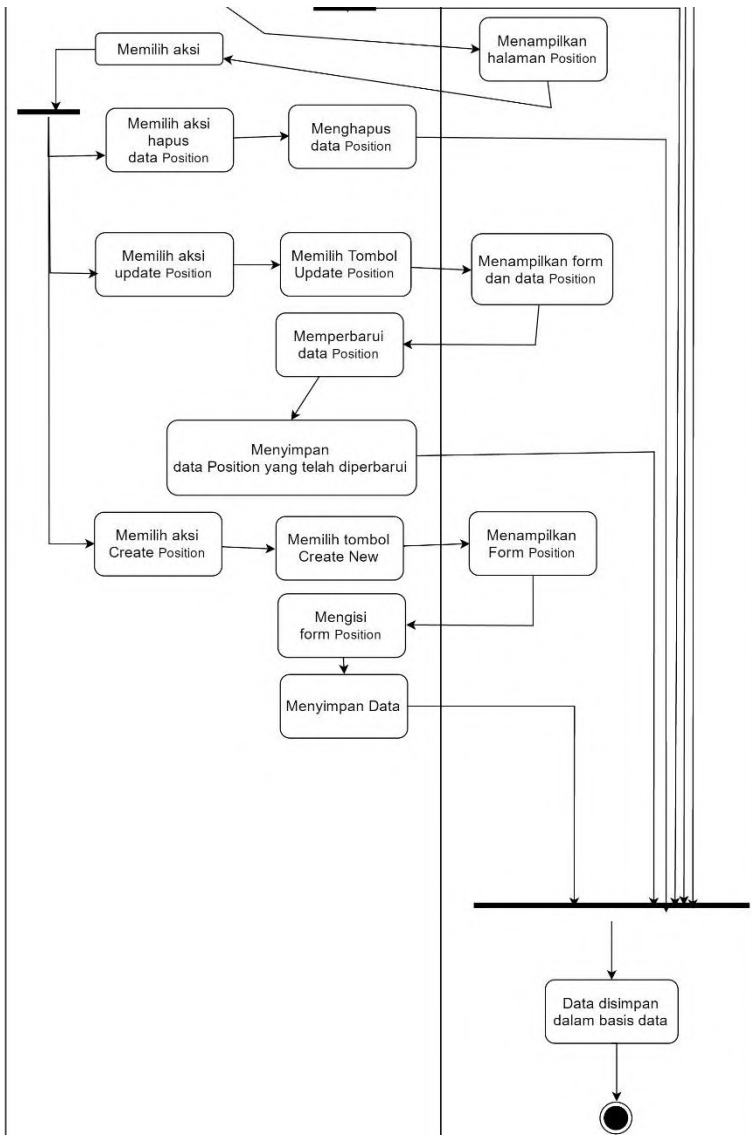
Tabel 3.7 Kasus Penggunaan Mengelola Pembuatan Struktur Kepegawaian

Nama	Mengelola Pembuatan Struktur Kepegawaian
Kode	UCTA-002
Deskripsi	Pengguna memilih menu <i>Employee Structure</i> . Menu ini digunakan untuk membuat struktur perusahaan mulai dari <i>Job</i> dan <i>Position</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan tombol “Create”
Aktor	Staff Human Resource Management
Kondisi Awal	
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan menu <i>Employee Structure</i> 2. Pengguna memilih menu <i>Job</i>. 3. Sistem menampilkan halaman <i>Job</i> 4. Pengguna mengisi form <i>Position</i>. 5. Pengguna menyimpan data <i>Job</i>. 6. Sistem menyimpan data <i>Job</i> ke dalam basis data. 7. Pengguna memilih menu <i>Position</i> 8. Sistem menampilkan halaman <i>Position</i>. 9. Pengguna memilih aksi <i>Create</i>. 10. Sistem menampilkan form <i>Position</i>. 11. Pengguna mengisi form <i>Position</i>.

	12. Pengguna menyimpan data <i>Position</i> . 13. Sistem menyimpan data <i>Position</i> ke dalam basis data.
- Kejadian Alternatif	2a. Pengguna memilih menu <i>Position</i> Sistem kembali ke kejadian normal 2. 7a. Pengguna memilih menu <i>Job</i> Sistem kembali ke kejadian normal 7.
Kondisi Akhir	Data struktur kepegawaian masuk ke dalam basis data.



Gambar 3.15 Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Kepegawaian Bagian (1)



Gambar 3.16 Diagram Aktivitas Kasus Penggunaan Mengelola Pembuatan Struktur Kpegawaian Bagian (2)

3.6.3 Kasus Penggunaan Mengelola Perekrutan Pegawai

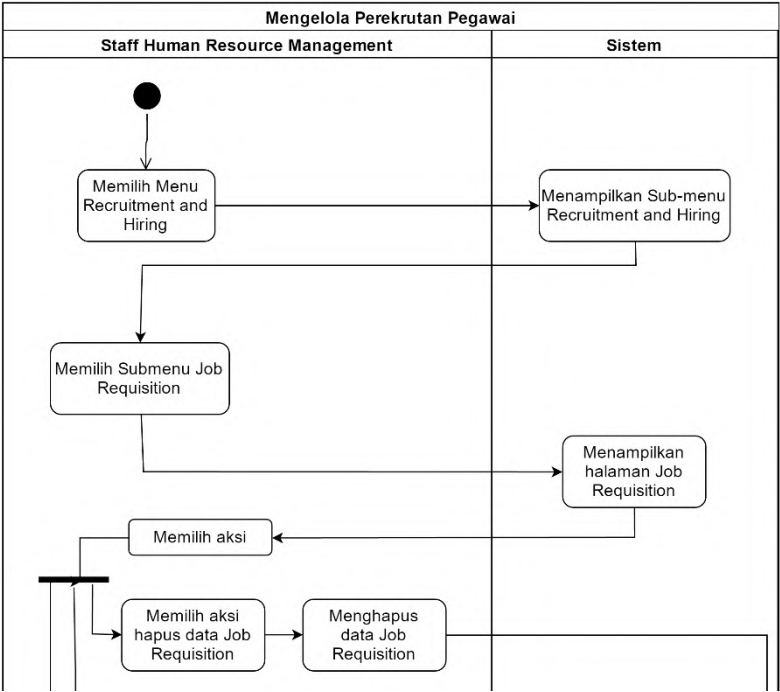
Pada kasus penggunaan ini, sistem menampilkan menu untuk melakukan perekrutan pegawai. Pengguna dapat melakukan proses perkrutan pegawai dengan mengisi form yang tersedia. Pengguna dapat menambah kandidat pegawai.

Tabel 3.8 Kasus Penggunaan Mengelola Perekrutan Pegawai

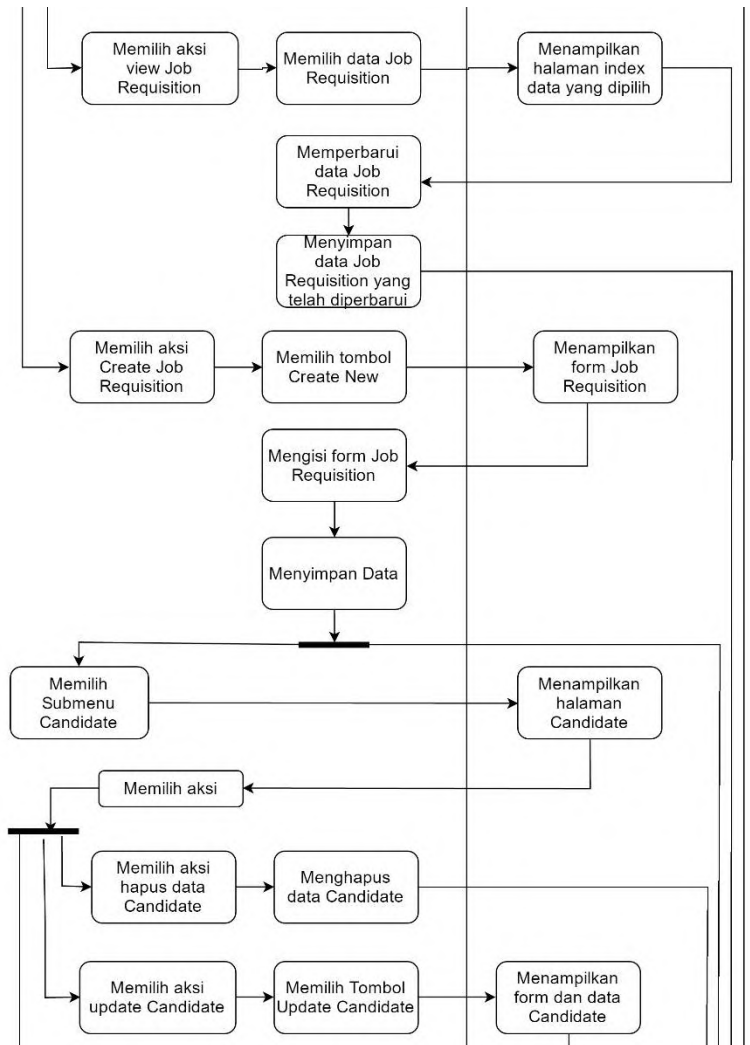
Nama	Mengelola Perekrutan Pegawai
Kode	UCTA-003
Deskripsi	Pengguna memilih menu <i>Recruitment and Hiring</i> . Lalu pengisian diurutkan dari <i>Job Requisition</i> , <i>Candidate</i> dan <i>Hire Candidate</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan tombol “Create”
Aktor	Staff Human Resource Management
Kondisi Awal	Ada status ‘open’ di salah satu Position.
Aliran: - Kejadian Normal	<ol style="list-style-type: none">1) Sistem menampilkan submenu ‘Recruitment and Hiring’.2) Pengguna memilih menu <i>Job Requisition</i>.3) Sistem menampilkan halaman <i>Job Requisition</i>.4) Pengguna menekan tombol <i>create new</i>.5) Sistem menampilkan form <i>job requisition</i>.6) Pengguna mengisi form <i>Job Requisition</i>.7) Pengguna menekan tombol <i>Create</i>8) Sistem menyimpan data ke dalam basis data.

	<p>9) Pengguna memilih menu <i>Candidate</i>.</p> <p>10) Sistem menampilkan halaman <i>Candidate</i>.</p> <p>11) Pengguna menekan tombol <i>Create</i>.</p> <p>12) Sistem menampilkan form <i>Candidate</i></p> <p>13) Pengguna mengisi form <i>candidate</i>.</p> <p>14) Pengguna menekan tombol <i>create</i>.</p> <p>15) Sistem menyimpan data ke dalam basis data.</p> <p>16) Pengguna memilih menu <i>Hire Candidate</i>.</p> <p>17) Sistem menampilkan halaman <i>Hire Candidate</i></p> <p>18) Pengguna menekan tombol <i>update</i>.</p> <p>19) Sistem menampilkan form <i>Hire Candidate</i>.</p> <p>20) Pengguna mengisi form <i>Hire candidate</i>.</p> <p>21) Pengguna menekan tombol <i>Save</i></p> <p>22) Sistem menyimpan data ke dalam basis data.</p>
- Kejadian Alternatif	<p>2a. Pengguna memilih menu <i>Candidate</i></p> <p>2b. Pengguna memilih menu <i>Hire Candidate</i></p> <p>Sistem kembali ke kejadian normal 2.</p> <p>9a. Pengguna memilih menu <i>Job Requisition</i></p>

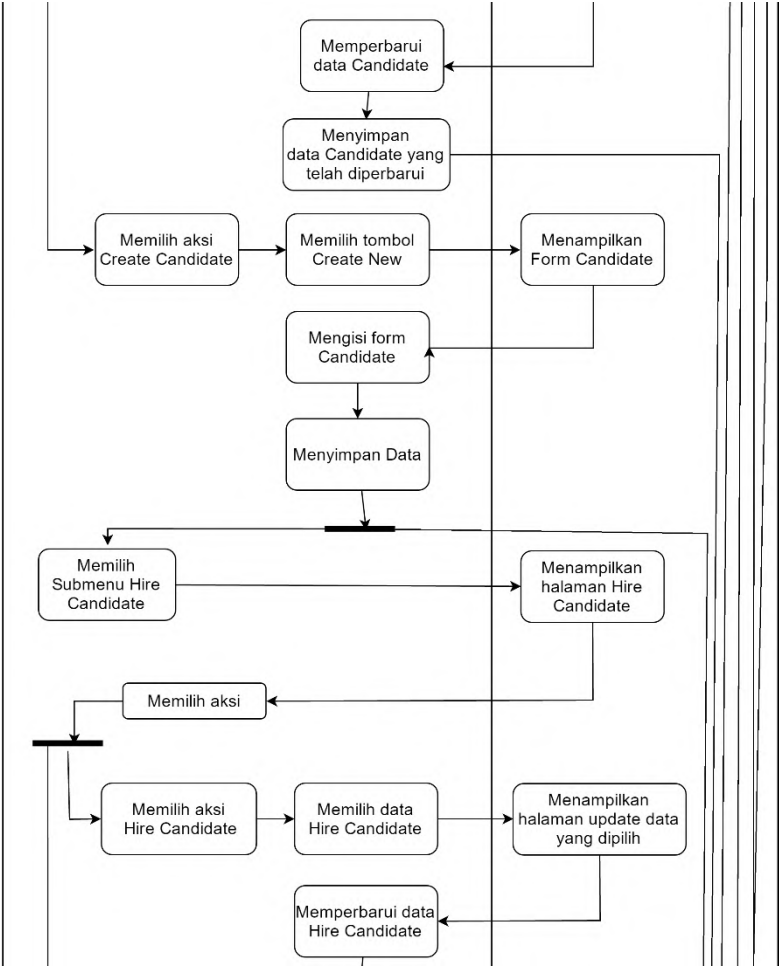
	<p>9b. Pengguna memilih menu <i>Hire Candidate</i></p> <p>Sistem kembali ke kejadian normal 9.</p> <p>16a. Pengguna memilih menu <i>Job Requisition</i></p> <p>16b. Pengguna memilih menu <i>Candidate</i></p> <p>Sistem kembali ke kejadian normal 16.</p>
Kondisi Akhir	Data perekrutan pegawai masuk ke dalam basis data.



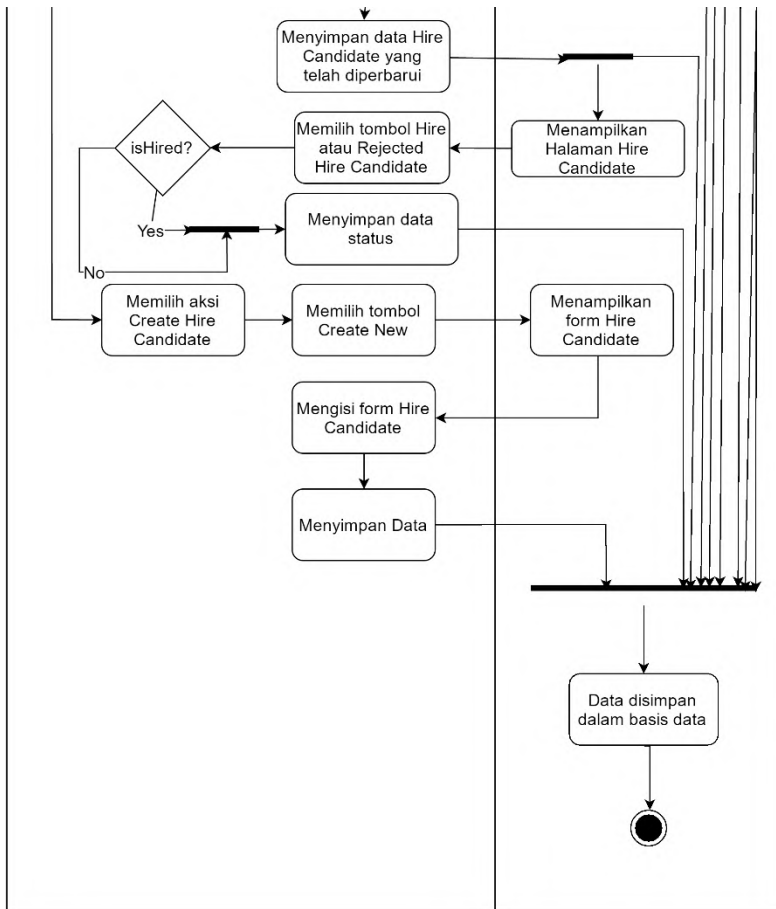
Gambar 3.17 Diagram Aktivitas Kasus Penggunaan Mengelola Perekrutan Pegawai Bagian (1)



Gambar 3.18 Diagram Aktivitas Kasus Penggunaan Perekrutan Pegawai Bagian (2)



Gambar 3.19 Diagram Aktivitas Kasus Penggunaan Perekrutan Pegawai Bagian (3)



Gambar 3.20 Diagram Aktivitas Kasus Penggunaan Perekrutan Pegawai Bagian (4)

3.6.4 Kasus Penggunaan Mengelola Penempatan Pegawai

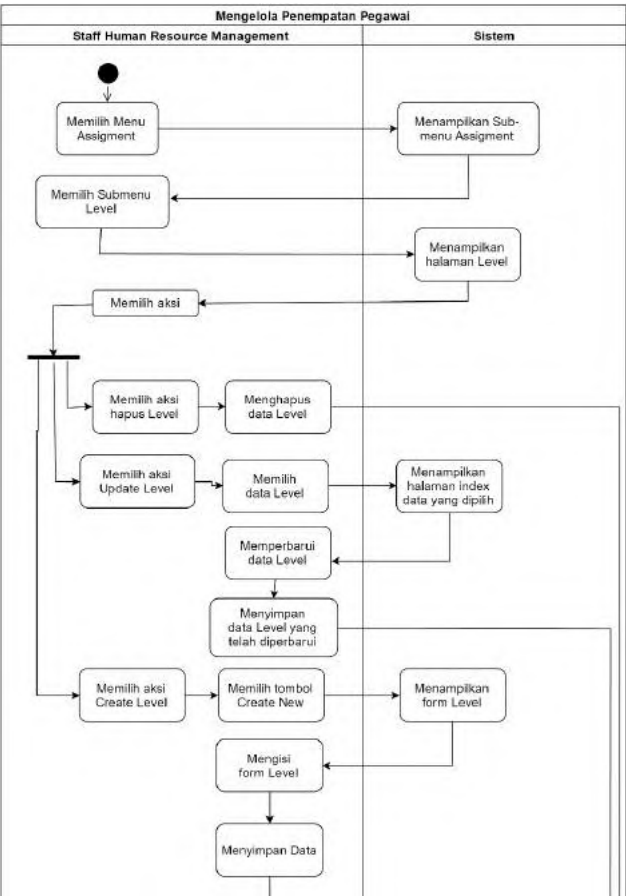
Pada kasus penggunaan ini, sistem menampilkan menu untuk melakukan penempatan pegawai. Pada kasus penggunaan ini pengguna melakukan proses administrasi kepegawaian. Pengguna dapat mencatat semua data pegawai yang dibutuhkan perusahaan.

Tabel 3.9 Kasus Penggunaan Mengelola Penempatan Pegawai

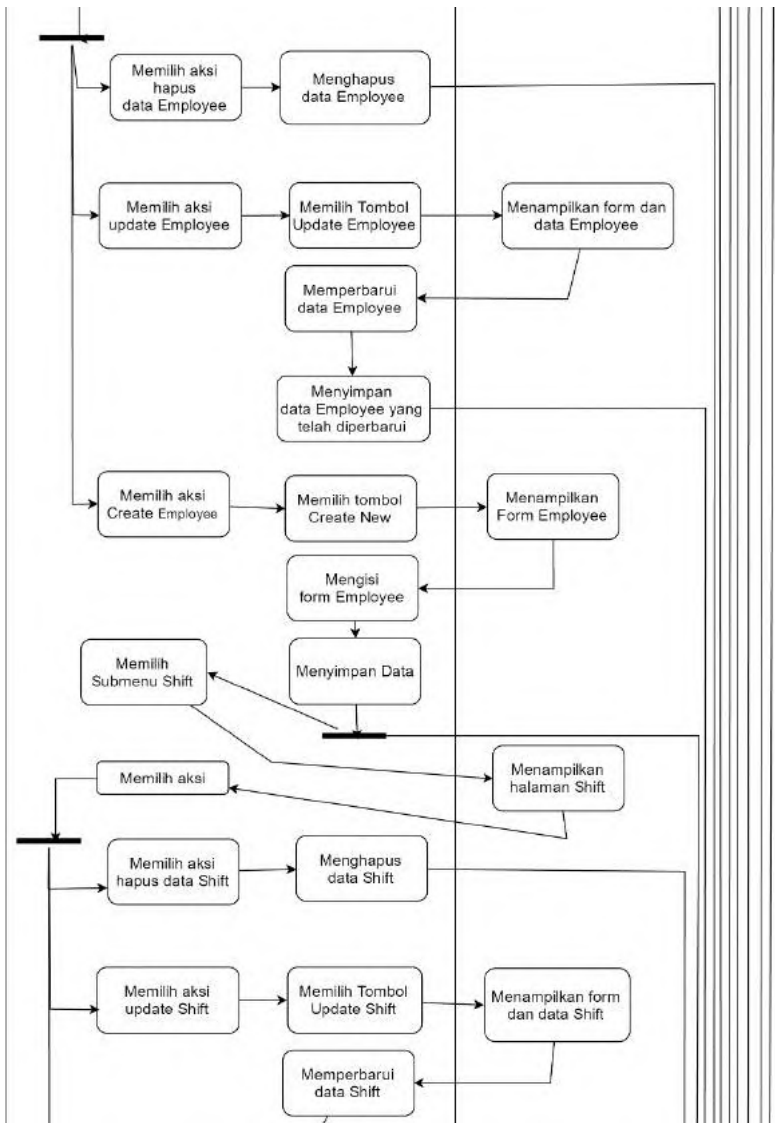
Nama	Mengelola Penempatan Pegawai
Kode	UCTA-004
Deskripsi	Pengguna memilih menu <i>Assignment</i> . Lalu membuat data <i>Level</i> , <i>Skill</i> , <i>Employee</i> dan <i>Shift</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan tombol “Create”
Aktor	Staff <i>Human Resource Management</i>
Kondisi Awal	Data <i>Personnel Sub Area</i> , <i>Skill</i> , dan <i>Position</i> sudah ada.
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan submenu ‘Recruitment and Hiring’. 2. Pengguna memilih menu <i>Level</i>. 3. Sistem menampilkan halaman <i>Level</i>. 4. Pengguna menekan tombol <i>create new</i>. 5. Sistem menampilkan form <i>Level</i> 6. Pengguna mengisi form <i>Level</i> 7. Pengguna menekan tombol <i>Create</i> 8. Sistem menyimpan data ke dalam basis data. 9. Pengguna memilih menu <i>Skill</i> 10. Sistem menampilkan halaman <i>Skill</i>. 11. Pengguna menekan tombol <i>Create</i>. 12. Sistem menampilkan form <i>Skill</i> 13. Pengguna mengisi form <i>Skill</i>. 14. Pengguna menekan tombol <i>create</i>. 15. Sistem menyimpan data ke dalam basis data.

	<p>16. Pengguna memilih menu <i>Employee</i></p> <p>17. Sistem menampilkan halaman <i>Employee</i></p> <p>18. Pengguna menekan tombol <i>update</i>.</p> <p>19. Sistem menampilkan form <i>Employee</i>.</p> <p>20. Pengguna mengisi form <i>Employee</i></p> <p>21. Pengguna menekan tombol <i>Save</i></p> <p>22. Sistem menyimpan data ke dalam basis data.</p> <p>23. Pengguna memilih menu <i>Shift</i></p> <p>24. Sistem menampilkan halaman <i>Shift</i>.</p> <p>25. Pengguna mengisi form <i>Shift</i></p> <p>26. Pengguna menekan tombol <i>Save</i>.</p> <p>27. Sistem menyimpan data ke dalam basis data</p>
- Kejadian Alternatif	<p>2a. Pengguna memilih menu <i>Skill</i></p> <p>2b. Pengguna memilih menu <i>Employee</i></p> <p>2c. Pengguna memilih menu <i>Shift</i></p> <p>Sistem kembali ke kejadian normal 2.</p> <p>9a. Pengguna memilih menu <i>Level</i></p> <p>9b. Pengguna memilih menu <i>Employee</i></p> <p>9c. Pengguna memilih menu <i>Shift</i></p> <p>Sistem kembali ke kejadian normal 9.</p> <p>16a. Pengguna memilih menu <i>Level</i></p> <p>16b. Pengguna memilih menu <i>Skill</i></p>

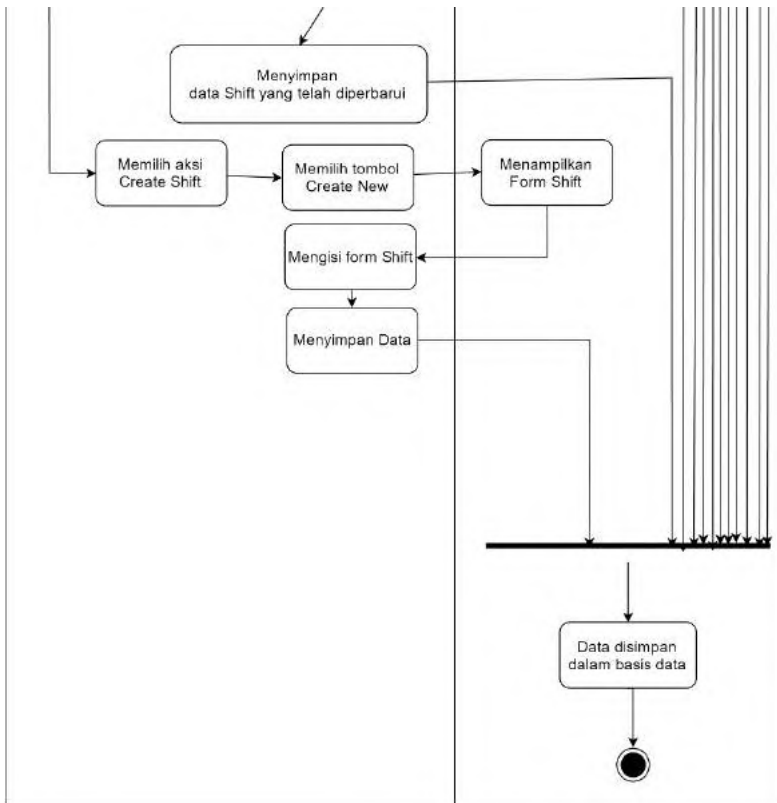
	16c. Pengguna memilih menu <i>Employee</i>
	Sistem kembali ke kejadian normal 16.
Kondisi Akhir	Data penempatan pegawai masuk ke dalam basis data.



Gambar 3.21 Diagram Aktivitas Kasus Penggunaan Penempatan Pegawai Bagian (1)



Gambar 3.23 Diagram Aktivitas Kasus Penggunaan Penempatan Pegawai Bagian (3)



Gambar 3.24 Diagram Aktivitas Kasus Penggunaan Penempatan Pegawai Bagian (4)

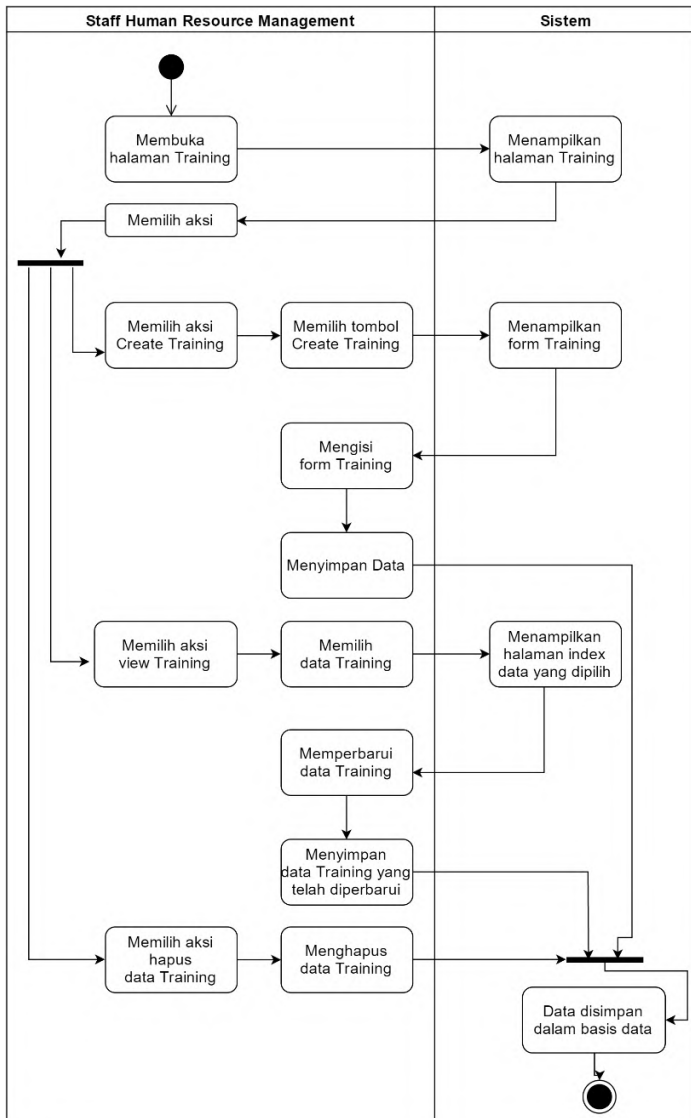
3.6.5 Kasus Penggunaan Mengelola Training

Pada kasus penggunaan ini, sistem menampilkan halaman Training. Training dilakukan terhadap Employee yang telah diproses sebelumnya pada (UCTA-003) dan (UCTA004).

Tabel 3.10 Kasus Penggunaan Mengelola Training

Nama	Mengelola Training
Kode	UCTA-005

Deskripsi	Sistem menampilkan halaman <i>Training</i> , kemudian sistem menerima masukan data training berdasarkan data Employee pada (UCTA-003). Sistem menyimpan masukan kedalam basis data.
Tipe	Fungsional
Pemicu	Pengguna menekan tombol “Create”
Aktor	Staff Human Resource Management
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>create Training</i>
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan menu <i>Training</i>. 2. Pengguna memilih menu <i>Training</i>. 3. Sistem menampilkan halaman <i>Training</i>. 4. Pengguna menekan tombol <i>create new</i>. 5. Sistem menampilkan form <i>Training</i> 6. Pengguna mengisi form <i>Training</i> 7. Pengguna menekan tombol <i>Create</i> 8. Sistem menyimpan data ke dalam basis data.
- Kejadian Alternatif	-
Kondisi Akhir	Data Training masuk ke dalam basis data.



Gambar 3.25 Diagram Aktivitas Kasus Penggunaan Mengelola Training

3.6.6 Kasus Penggunaan Mengelola Bisnis Goal

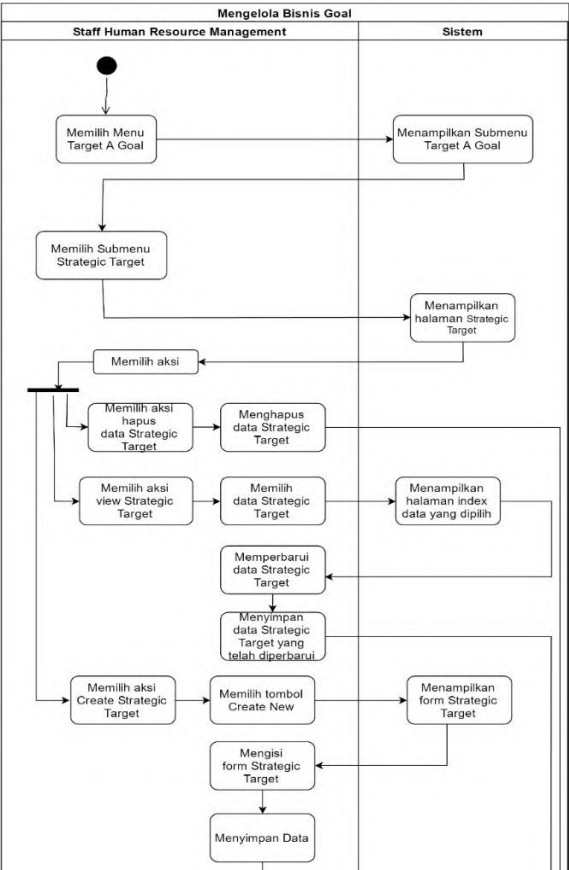
Pada kasus penggunaan ini, sistem menampilkan menu untuk mendeskripsikan bisnis goal perusahaan. Hasil dari bisnis goal ini untuk menentukan besar bonus untuk setiap karyawan.

Tabel 3.11 Kasus Penggunaan Mengelola Bisnis Goal

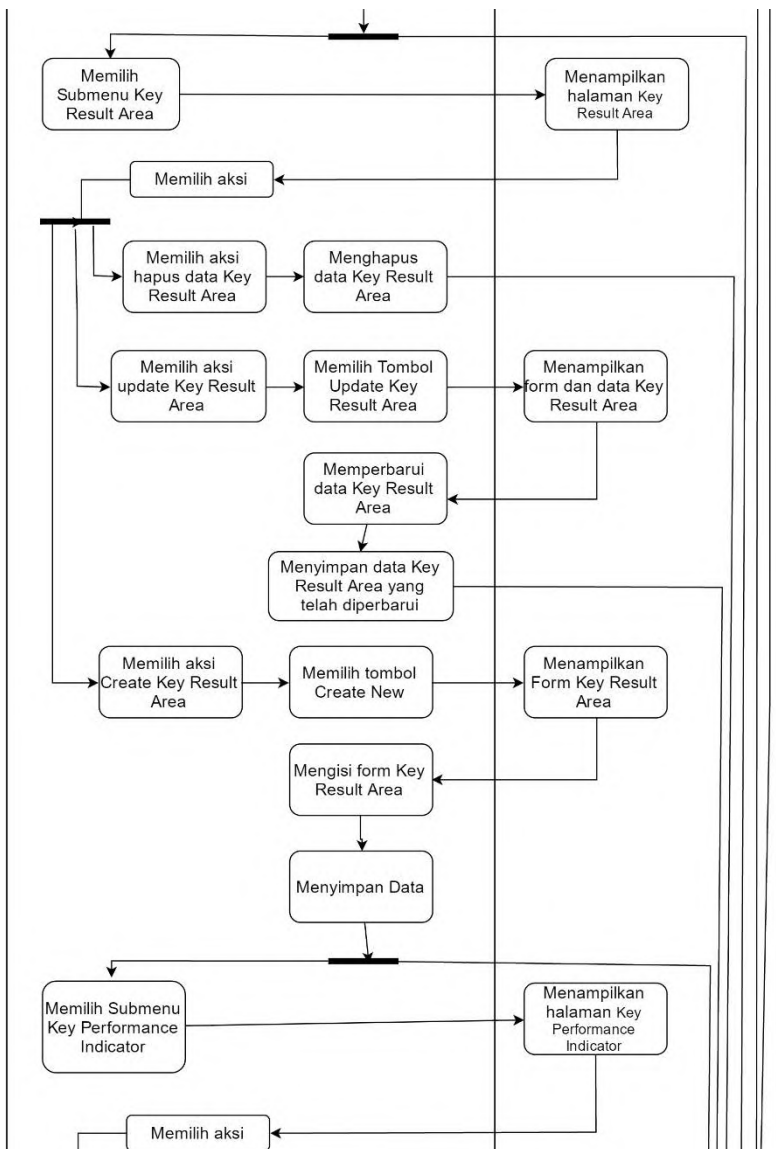
Nama	Mengelola Bisnis Goal
Kode	UCTA-006
Deskripsi	Pengguna memilih menu <i>Target A Goal</i> . Lalu pengisian diurutkan dari <i>Strategic Target</i> , <i>Key Result Area</i> dan <i>Key Performance Indicator</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan tombol “Create”
Aktor	Staff Human Resource Management
Kondisi Awal	Sistem menampilkan menu <i>Target A Goal</i>
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan submenu ‘Target A Goal’. 2. Pengguna memilih menu <i>Strategic Target</i>. 3. Sistem menampilkan halaman <i>Strategic Target</i>. 4. Pengguna menekan tombol <i>create new</i>. 5. Sistem menampilkan form <i>Strategic Target</i>. 6. Pengguna mengisi form <i>Strategic Target</i>. 7. Pengguna menekan tombol <i>Create</i> 8. Sistem menyimpan data ke dalam basis data. 9. Pengguna memilih menu <i>Key Result Area</i>

	<p>10. Sistem menampilkan halaman <i>Key Result Area</i></p> <p>11. Pengguna menekan tombol <i>Create</i>.</p> <p>12. Sistem menampilkan form <i>Key Result Area</i></p> <p>13. Pengguna mengisi form <i>Key Result Area</i>.</p> <p>14. Pengguna menekan tombol <i>create</i>.</p> <p>15. Sistem menyimpan data ke dalam basis data.</p> <p>16. Pengguna memilih menu <i>Key Performance Indicator</i></p> <p>17. Sistem menampilkan halaman <i>Key Performance Indicator</i></p> <p>18. Pengguna menekan tombol <i>update</i>.</p> <p>19. Sistem menampilkan form <i>Key Performance Indicator</i>.</p> <p>20. Pengguna mengisi form <i>Key Performance Indicator</i></p> <p>21. Pengguna menekan tombol <i>Save</i></p> <p>22. Sistem menyimpan data ke dalam basis data.</p>
- Kejadian Alternatif	<p>2a. Pengguna memilih menu <i>Key Result Area</i></p> <p>2b. Pengguna memilih menu <i>Key Performance Indicator</i></p> <p>Sistem kembali ke kejadian normal 2.</p> <p>9a. Pengguna memilih menu <i>Strategic Target</i></p> <p>9b. Pengguna memilih menu <i>Key Performance Indicator</i></p> <p>Sistem kembali ke kejadian normal 9.</p>

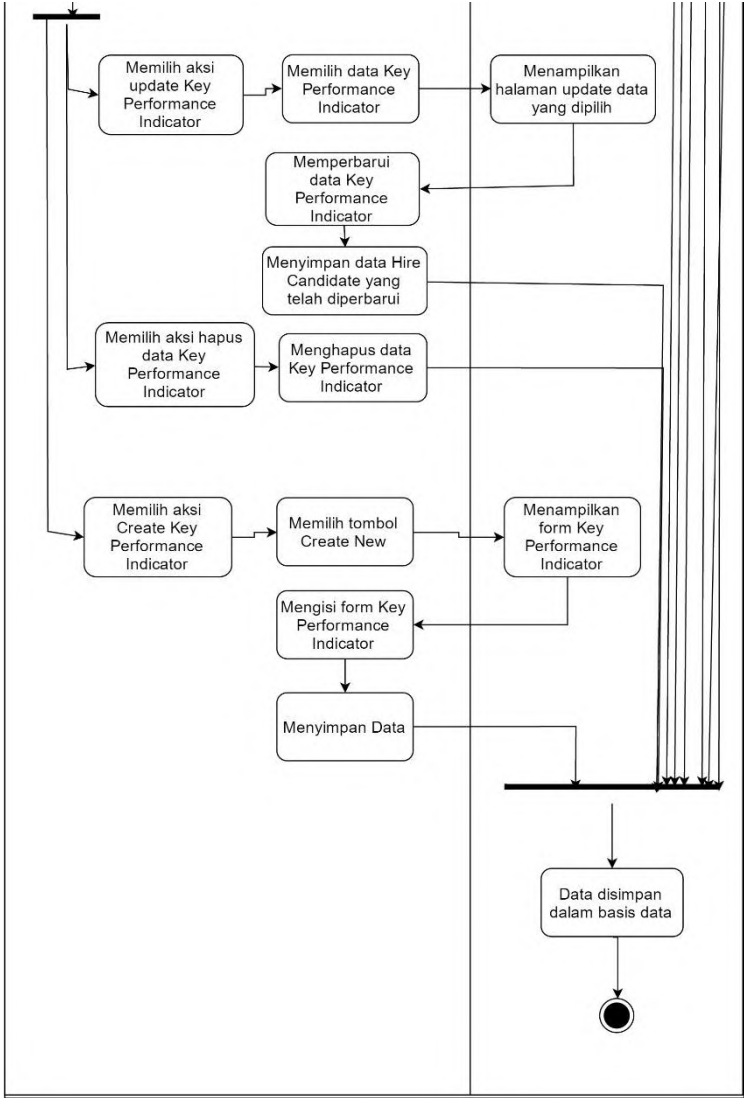
	16a. Pengguna memilih menu <i>Strategic Target</i> 16b. Pengguna memilih menu <i>Key Result Area</i> Sistem kembali ke kejadian normal 16.
Kondisi Akhir	Data Bisnis Goal masuk ke dalam basis data.



Gambar 3.26 Diagram Aktivitas Kasus Penggunaan Mengelola Bisnis Goal Bagian (1)



Gambar 3.27 Diagram Aktivitas Kasus Penggunaan Mengelola Bisnis Goal Bagian (2)



Gambar 3.28 Diagram Aktivitas Kasus Penggunaan Mengelola Bisnis Goal Bagian (3)

3.6.7 Kasus Penggunaan Mengelola Penggajian Pegawai

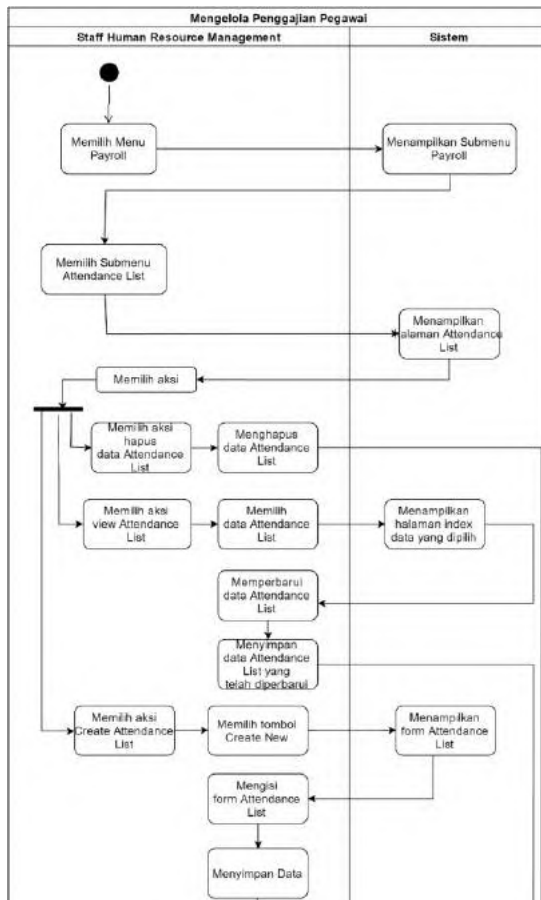
Pada kasus penggunaan ini, sistem menampilkan menu untuk melakukan penggajian pegawai. Sebelum proses penghitungan gaji secara keseluruhan, pengguna dapat memasukkan nilai tunjangan dan potongan yang ada di perusahaan.

Tabel 3.12 Kasus Penggunaan Mengelola Penggajian Pegawai

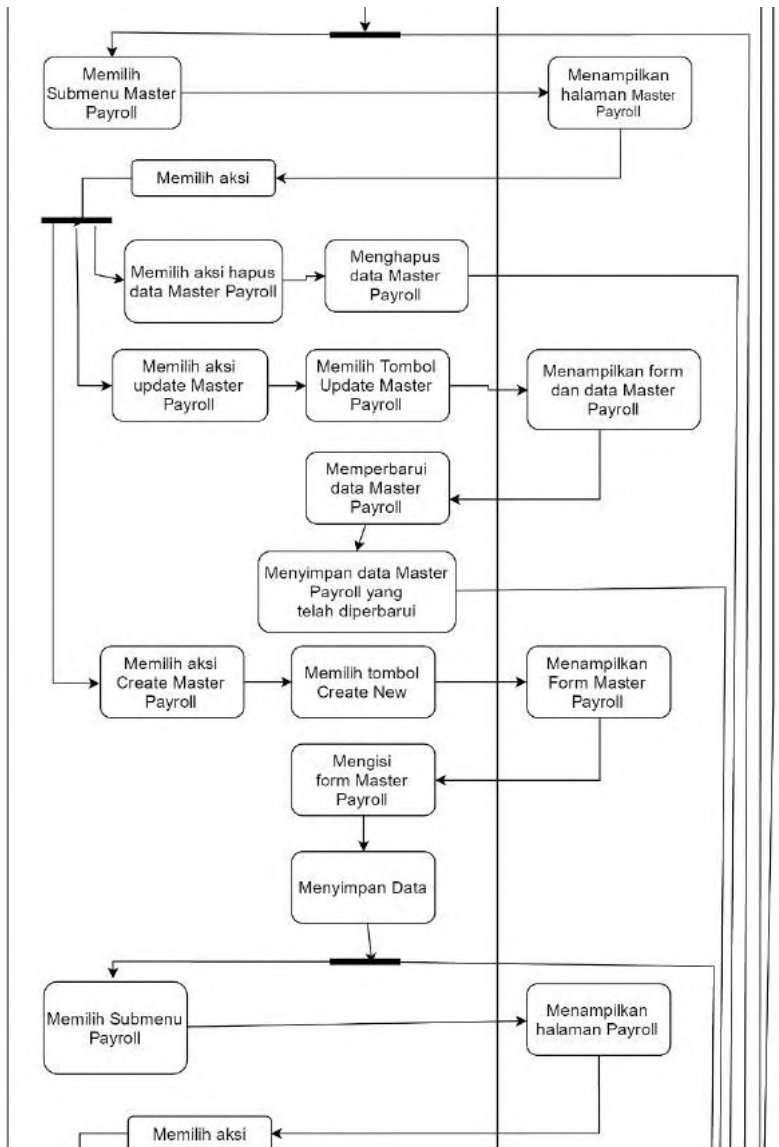
Nama	Mengelola Penggajian
Kode	UCTA-007
Deskripsi	Pengguna memilih menu <i>Payroll</i> . Lalu pengisian diurutkan dari <i>Attendande List</i> , <i>Master Payroll</i> dan <i>Payroll</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan tombol “Create”
Aktor	Staff Human Resource Management
Kondisi Awal	Sistem menampilkan menu <i>Payroll</i> .
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1) Sistem menampilkan submenu ‘Payroll’. 2) Pengguna memilih menu <i>Attendance List</i>. 3) Sistem menampilkan halaman <i>Attendance List</i>. 4) Pengguna menekan tombol <i>create new</i>. 5) Sistem menampilkan form <i>Attendance List</i>. 6) Pengguna mengisi form <i>Attendance List</i> 7) Pengguna menekan tombol <i>Create</i> 8) Sistem menyimpan data ke dalam basis data. 9) Pengguna memilih menu <i>Master Payroll</i>

	<p>10) Sistem menampilkan halaman <i>Master Payroll</i></p> <p>11) Pengguna menekan tombol <i>Create</i>.</p> <p>12) Sistem menampilkan form <i>Master Payroll</i></p> <p>13) Pengguna mengisi form <i>Master Payroll</i></p> <p>14) Pengguna menekan tombol <i>create</i>.</p> <p>15) Sistem menyimpan data ke dalam basis data.</p> <p>16) Pengguna memilih menu <i>Payroll</i></p> <p>17) Sistem menampilkan halaman <i>Payroll</i></p> <p>18) Pengguna menekan tombol <i>update</i>.</p> <p>19) Sistem menampilkan form <i>Payroll</i></p> <p>20) Pengguna mengisi form <i>Payroll</i></p> <p>21) Pengguna menekan tombol <i>Save</i></p> <p>22) Sistem menyimpan data ke dalam basis data.</p>
- Kejadian Alternatif	<p>2a. Pengguna memilih menu <i>Master Payroll</i></p> <p>2b. Pengguna memilih menu <i>Payroll</i> Sistem kembali ke kejadian normal 2.</p> <p>9a. Pengguna memilih menu <i>Attendance List</i></p> <p>9b. Pengguna memilih menu <i>Payroll</i> Sistem kembali ke kejadian normal 9.</p> <p>16a. Pengguna memilih menu <i>Attendance List</i></p>

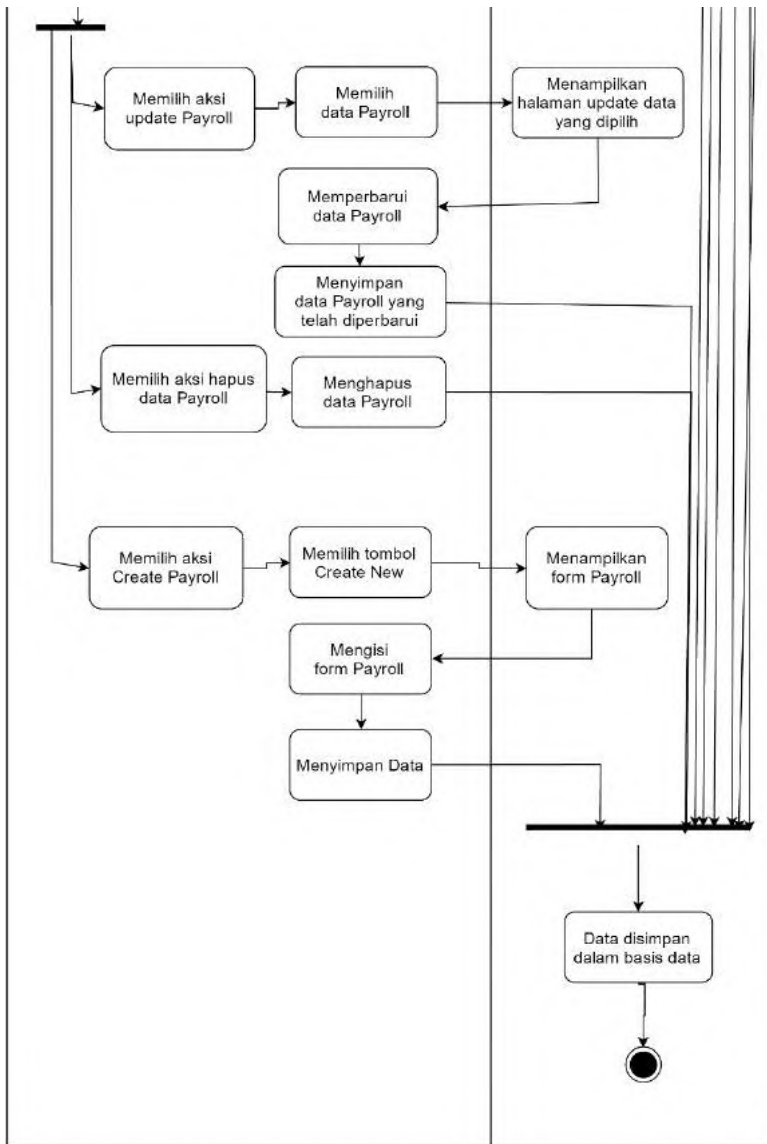
	16b. Pengguna memilih menu <i>Master Payroll</i>
	Sistem kembali ke kejadian normal 16.
Kondisi Akhir	Data penggajian pegawai masuk ke dalam basis data.



Gambar 3.29 Diagram Aktivitas Kasus Penggunaan Mengelola Penggajian Pegawai Bagian (1)



Gambar 3.30 Diagram Aktivitas Kasus Penggunaan Mengelola Penggajian Pegawai Bagian (2)



Gambar 3.31 Diagram Aktivitas Kasus Penggunaan Mengelola Penggajian Pegawai Bagian (3)

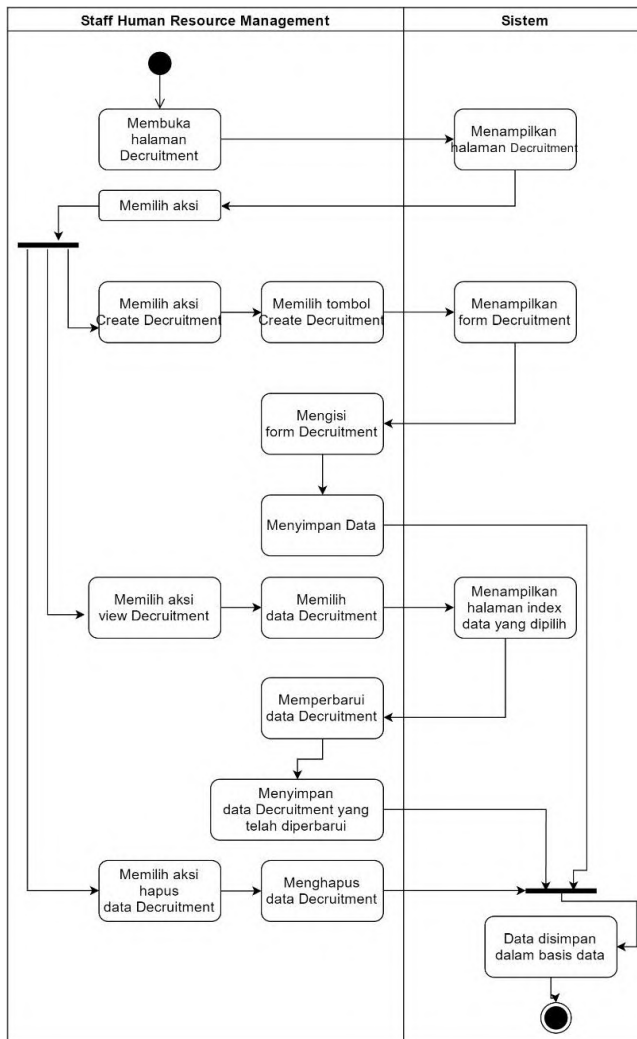
3.6.8 Kasus Penggunaan Mengelola Pengurangan Pegawai

Pada kasus penggunaan ini, sistem menampilkan menu untuk melakukan pengurangan pegawai. Pengguna dapat mengurangi data pegawai melalui kasus penggunaan ini. Sistem juga akan mencatat alasan setiap pegawai yang berkurang.

Tabel 3.13 Kasus Penggunaan Mengelola Pengurangan Pegawai

Nama	Mengelola Pengurangan Pegawai
Kode	UCTA-008
Deskripsi	Sistem menampilkan halaman view <i>Decruitment</i>
Tipe	Fungsional
Pemicu	Pengguna menekan menu <i>Decruitment</i>
Aktor	Staff Human Resource Management
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Decruitment</i>
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan menu <i>Decruitment</i>. 2. Pengguna memilih menu <i>Decruitment</i>. 3. Sistem menampilkan halaman <i>Decruitment</i>. 4. Pengguna menekan tombol <i>create new</i>. 5. Sistem menampilkan form <i>Decruitment</i> 6. Pengguna mengisi form <i>Decruitment</i> 7. Pengguna menekan tombol <i>Create</i> 8. Sistem menyimpan data ke dalam basis data.
- Kejadian Alternatif	-

Kondisi Akhir	Data pengurangan pegawai masuk kedalam basis data
----------------------	--



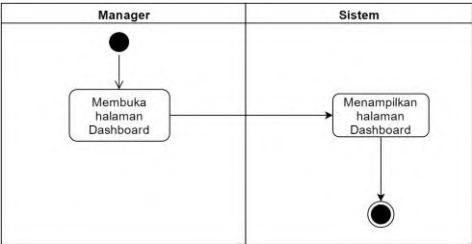
Gambar 3.32 Diagram Aktivitas Kasus Penggunaan Mengelola Pengurangan Pegawai

3.6.9 Kasus Penggunaan Melihat Halaman Dashboard

Pada kasus penggunaan ini, sistem menampilkan menu untuk melakukan perekrutan pegawai. Pengguna dapat melakukan proses perekrutan pegawai dengan mengisi form yang tersedia. Pengguna dapat menambah kandidat pegawai.

Tabel 3.14 Kasus Penggunaan Melihat Halaman Dashboard

Nama	Melihat Halaman Dashboard
Kode	UCTA-009
Deskripsi	Pengguna memilih menu <i>Dashboard</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan menu <i>Dashboard</i>
Aktor	Manager Human Resource Management
Kondisi Awal	Sistem menampilkan menu <i>Dashboard</i>
Aliran: - Kejadian Normal	1. Sistem menampilkan menu <i>Dashabord</i> . 2. Pengguna memilih menu <i>Dashboard</i> . 3. Sistem menampilkan halaman <i>Dashboard</i> .
- Kejadian Alternatif	-
Kondisi Akhir	Data dapat dilihat oleh pengguna



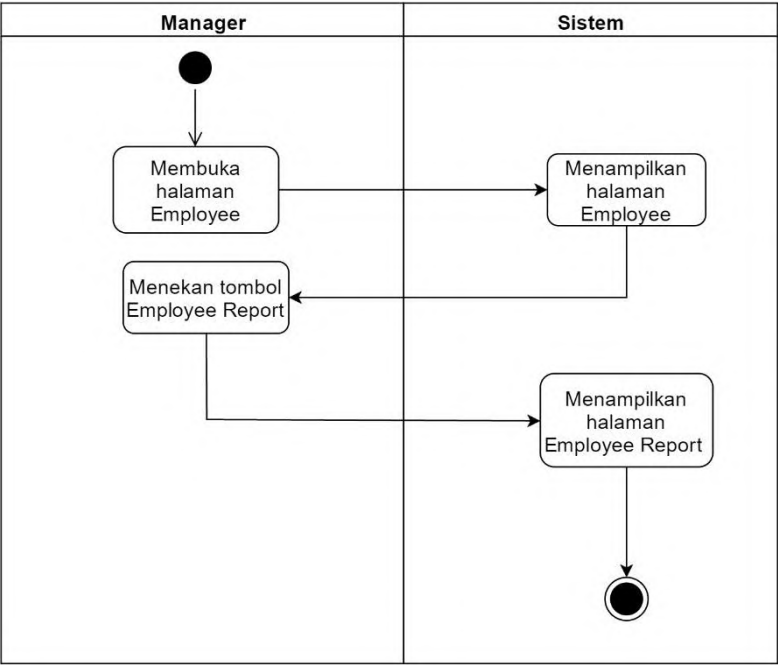
Gambar 3.33 Diagram Aktivitas Kasus Penggunaan Melihat Halaman Dashboard

3.6.10 Kasus Penggunaan Melihat Laporan Pegawai

Pada kasus penggunaan ini, sistem menampilkan menu untuk laporan pegawai. Halaman ini digunakan untuk melihat data pegawai secara keseluruhan. Laporan ini menampilkan hasil informasi dari data pegawai yang ada di dalam perusahaan. Mulai dari jumlah, jenis, bidang keahlian dan lain lain.

Tabel 3.15 Kasus Penggunaan Melihat Laporan Pegawai

Nama	Melihat Halaman Laporan
Kode	UCTA-010
Deskripsi	Pengguna menekan tombol <i>Employee Report</i> .
Tipe	Fungsional
Pemicu	Pengguna menekan menu <i>Employee Report</i>
Aktor	Manager Human Resource Management
Kondisi Awal	Sistem menampilkan menampilkan halaman <i>Employee</i>
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan tombol <i>Employee Report</i>. 2. Pengguna menekan Tombol <i>Employee Report</i>. 3. Sistem menampilkan halaman <i>Employee Report</i>.
- Kejadian Alternatif	-
Kondisi Akhir	Data dapat dilihat oleh pengguna



Gambar 3.34 Diagram Aktivitas Kasus Penggunaan Melihat Laporan Pegawai

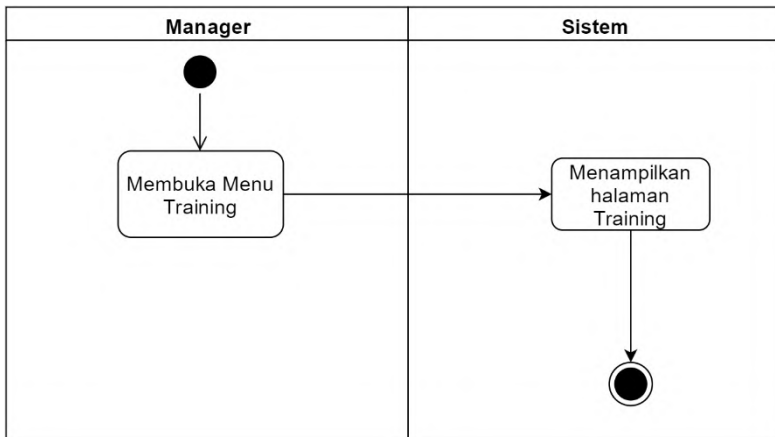
3.6.11 Kasus Penggunaan Melihat Data *Training*

Pada kasus penggunaan ini, sistem menampilkan halaman *Training*. Dimana pada halaman ini ditampilkan daftar *training* yang ada di dalam perusahaan.

Tabel 3.16 Kasus Penggunaan Melihat Data *Training*

Nama	Melihat Halaman Laporan
Kode	UCTA-011
Deskripsi	Pengguna memilih menu <i>Training</i> . Maka sistem akan menampilkan daftar <i>Training</i> yang ada.
Tipe	Fungsional
Pemicu	Pengguna menekan menu <i>Training</i>

Aktor	Manager Human Resource Management
Kondisi Awal	Sistem menampilkan menampilkan halaman <i>Training</i>
Aliran: - Kejadian Normal	1. Pengguna memilih menu Training 2. Sistem menampilkan halaman <i>Training</i> .
- Kejadian Alternatif	-
Kondisi Akhir	Data dapat dilihat oleh pengguna



Gambar 3.35 Diagram Aktivitas Kasus Penggunaan Melihat Data *Training*

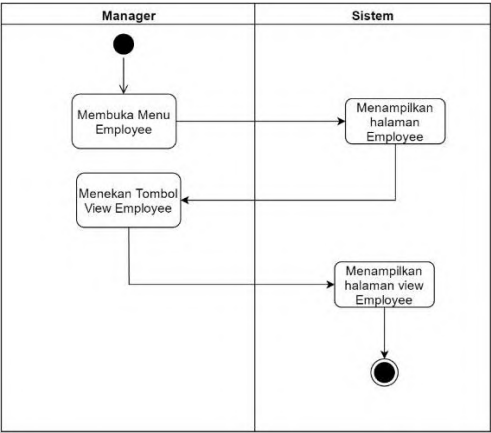
3.6.12 Kasus Penggunaan Melihat Data Pegawai

Pada kasus penggunaan ini, sistem menampilkan halaman *Employee*. Dimana pada halaman ini ditampilkan daftar *Employee* yang ada di dalam perusahaan.

Tabel 3.17 Kasus Penggunaan Melihat Data Pegawai

Nama	Melihat Data Pegawai
Kode	UCTA-012

Deskripsi	Pengguna memilih menu Training. Maka sistem akan menampilkan daftar <i>Training</i> yang ada.
Tipe	Fungsional
Pemicu	Pengguna menekan menu <i>Employee</i>
Aktor	Manager Human Resource Management
Kondisi Awal	Sistem menampilkan menampilkan halaman <i>Employee</i>
Aliran: - Kejadian Normal	1. Pengguna memilih menu <i>Employee</i> 2. Sistem menampilkan halaman <i>Employee</i> . 3. Pengguna menekan tombol <i>View Employee</i> . 4. Sistem menampilkan halaman <i>View</i> .
- Kejadian Alternatif	-
Kondisi Akhir	Data dapat dilihat oleh pengguna



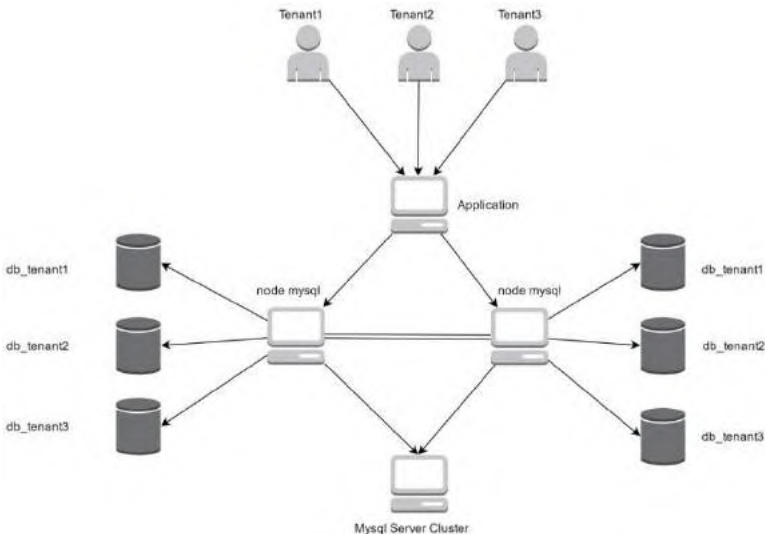
Gambar 3.36 Diagram Aktivitas Kasus Penggunaan Melihat Data Pegawai

3.7 Perancangan Sistem

Tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan arsitektur basis data terdistribusi, perancangan *multi-tenancy*, perancangan RBAC, dan perancangan antarmuka sistem.

3.7.1 Perancangan *Multitenancy*

Pada bagian ini akan dijelaskan mengenai perancangan *multitenancy* yang terdapat pada sistem ERP 2016. Secara rinci mengenai *multitenancy* dijabarkan pada Gambar 3.30.



Gambar 3.37 Perancangan *Multitenancy*

Gambar 3.28 menunjukkan alur *multitenancy* yang terjadi pada sistem ERP. Pembagian *node MySQL* yang telah dijelaskan pada subbab 3.6.1, yaitu dengan cara melakukan *clustering* pada *computer server* menjadi 2 basis data yang berbeda. Basis data terdistribusi merupakan penunjang dari *multitenancy*, yang berfungsi untuk mengatur *session* setiap perusahaan yang akan mengakses sistem ERP.

3.7.2 Perancangan RBAC

Pada bagian ini akan dijelaskan mengenai perancangan RBAC yang terdapat pada sistem ERP 2016. Perincian mengenai RBAC dijabarkan sebagai berikut:

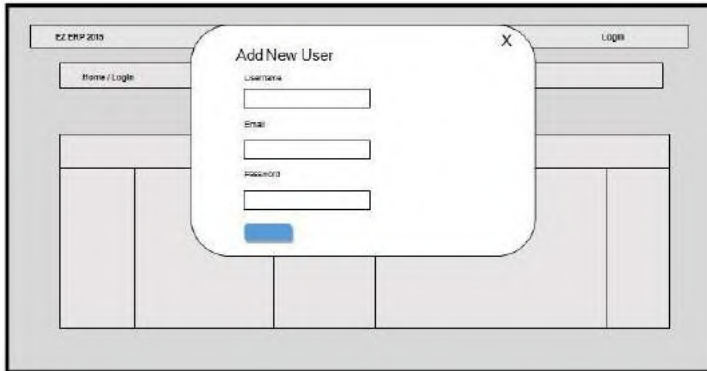
3.7.2.1 Perancangan Antarmuka Login

Rancangan antarmuka login dibagi menjadi 3 (tiga) bagian, di bagian paling atas terdapat header yang berisi logo aplikasi dan tombol login, di bagian atas terdapat navigasi (*breadcrumb*) untuk mengetahui user sedang berada pada halaman tertentu, dan di bagian tengah terdapat form login yang memiliki *username*, *password*, dan pemilihan basis data yang digunakan sebagai input. Perancangan antarmuka login ditunjukkan pada Gambar 3.31.

Gambar 3.38. Rancang antarmuka login

3.7.2.2 Perancangan Antarmuka Menambahkan User Baru

Rancangan antarmuka menambahkan user baru ditampilkan dalam sebuah kotak dialog yang berisi *username*, *email*, dan *password* yang digunakan sebagai input data untuk menambahkan user baru. Perancangna antarmuka menambahkan user baru ditunjukkan pada Gambar 3.32.



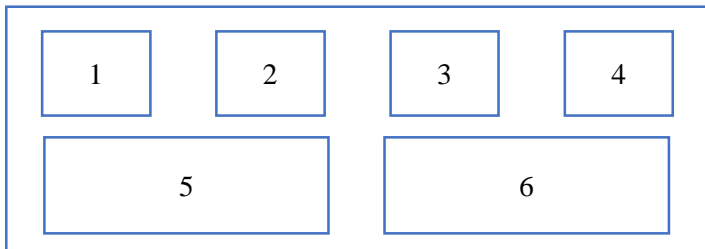
Gambar 3.39. Rancang Antarmuka Menambahkan Pengguna Baru

3.7.3 Perancangan Tampilan

Pada sub-bab ini membahas perancangan antarmuka yang akan digunakan dalam aplikasi ERP khususnya modul HRM. Ilustrasi perancangan ditunjukkan sebagai gambar disertai keterangan objek-objek yang ada di dalamnya.

3.7.3.1 Perancangan Antarmuka Halaman *Dashboard*

Rancangan antarmuka halaman dashboard ditunjukkan oleh Gambar 3.33. Pada halaman ini ditujukan untuk pengguna agar dapat memonitor keseluruhan modul HRM.



Gambar 3.40 Rancangan Antarmuka Halaman *Dashboard*

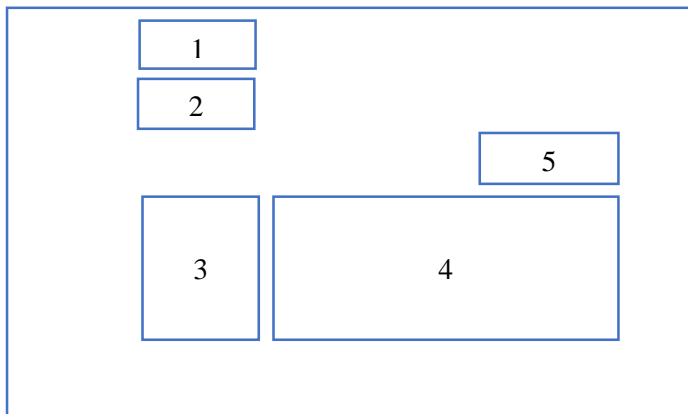
Berikut penjelasan masing-masing nomor yang ada di dalam masing-masing kotak:

1. Total *Direct* dan *Indirect* Labour di perusahaan.
2. *Turnover Rate* Pegawai.

3. Total KPI.
4. Total Gaji untuk *Direct* dan *Indirect* labour.
5. Daftar *Training* yang ada di dalam perusahaan.
6. Target dan pencapaiannya.

3.7.3.2 Perancangan Antarmuka Halaman Utama Submodul

Rancangan antarmuka halaman utama submodul ditunjukkan oleh Gambar 3.41. Pada halaman ini pengguna melihat halaman indeks atau utama dari setiap submodul.



Gambar 3.41 Rancangan Antarmuka Halaman Utama Submodul

Berikut penjelasan masing-masing nomor yang ada di dalam masing-masing kotak:

1. Teks bertuliskan “Nama Submodul”.
2. Tombol *Create*.
Jika tombol ini ditekan maka aplikasi akan mengarahkan ke halaman form tiap submodul.
3. Tombol *Action* untuk setiap data antara lain: *View*, *Update*, *Delete*.
4. Daftar data dari submodul.

5. Tombol untuk ekspor daftar data dari submodul.

3.7.3.3 Perancangan Antarmuka Halaman Form *Create dan Update*

Rancangan antarmuka halaman *View* Submodul ditunjukkan oleh Gambar 3.35. Pada halaman ini, pengguna dapat melihat detail dari setiap data yang ada di setiap submodul.

The diagram illustrates the layout of the 'View' Submodule Form. It consists of a large central rectangular area labeled '2'. Above this area is a small rectangular box labeled '1'. Below the central area are two smaller rectangular boxes labeled '3' and '4' positioned side-by-side.

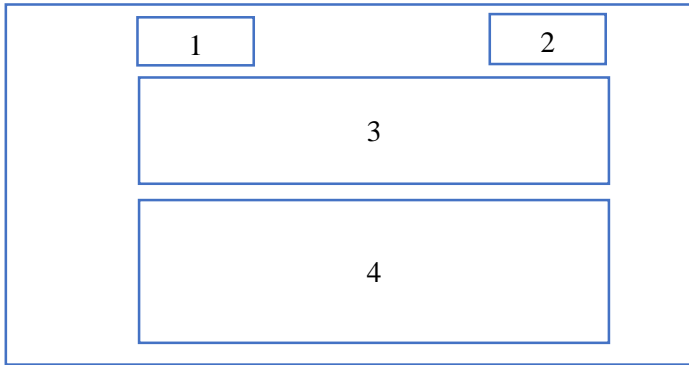
Gambar 3.42 Rancangan Antarmuka Halaman Form

Berikut penjelasan masing-masing nomor yang ada di dalam masing-masing kotak:

1. Teks bertuliskan “Nama Submodul”.
2. Form sesuai Submodul yang akan dibuat atau disunting.
3. Tombol Create atau Tombol Update data submodul.
4. Tombol Reset untuk reset data pada form.

3.7.3.4 Perancangan Antarmuka Halaman *View* Submodul

Rancangan antarmuka halaman *View* Submodul ditunjukkan oleh Gambar 3.42. Pada halaman ini, pengguna dapat melihat detail dari setiap data yang ada di setiap submodul.



Gambar 3.43 Rancangan Antarmuka Halaman View Submodul

Berikut penjelasan masing-masing nomor yang ada didalam masing-masing kotak:

1. Teks bertuliskan “Nama Submodul”.
2. Tombol *Action* untuk setiap data antara lain: *Pdf*, *Update*, *Delete*.
3. Detail setiap data yang ada di setiap submodul
4. Daftar data yang memiliki relasi terhadap submodul tersebut

BAB IV

IMPLEMENTASI SISTEM

Bab ini membahas implementasi dari perancangan sistem ERP 2016. Di dalamnya mencakup penjelasan lingkungan pengembangan sistem serta proses implementasi *distributed database*, RBAC, *multi-tenancy*, dan antarmuka pengguna.

4.1 Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan dan kaskas sebagai berikut.

1. Basis data yang digunakan pada *server* adalah MySQL Cluster.
2. 2 PC untuk *server database* menggunakan Sistem Operasi Ubuntu 12.04.
3. PC untuk *server* menggunakan Intel® Core™ i3-2120 @3.30GHz , RAM 4GB dengan Sistem Operasi Windows 8.1 Enterprise x64.
4. Mozilla Firefox 46.0.1 dan Chrome 49.0 sebagai antarmuka untuk pengujian aplikasi klien

4.2 Implementasi Basis Data Terdistribusi

Pada bagian ini akan dijelaskan mengenai implementasi *distributed database* yang terdapat pada sistem ERP 2016. Secara rinci mengenai implementasi *distributed database* dijabarkan sebagai berikut:

4.2.1 Instalasi Data dan SQL node pada node1 dan node2

Pada implementasi instalasi dan sql node pada node1 dan node2, dilakukan langkah-langkah sebagai berikut:

1. Membuat grup MySQL pengguna baru, kemudian menambah user MySQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.1.

```
shell> groupadd mysql
shell> useradd-g mysql mysql
```

Kode Sumber 4.1 Membuat grup MySQL pengguna baru dan menambah user MySQL

2. Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, kemudian mengubah arsip dan menciptakan symlink ke dalam direktori mysql yang bernama “mysql”. Hal yang perlu diperhatikan adalah, file yang sebenarnya dan nama direktori bervariasi sesuai dengan jumlah cluster versi MySQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.2.

```
shell> cd/usr/local
shell>/usr/local$ tar xzvf mysql-cluster-gpl-
7.1.34-linux-x86_64-glibc23
shell>ln-s/usr/local/mysql-cluster-gpl-7.1.34-
linux-x86-glibc23/usr/local/mysql
shell> export PATH = $ PATH:/usr/local/mysql/bin
shell> echo "export PATH=$PATH:/usr/local/
mysql/bin">> /etc/bash.bashrc
```

Kode Sumber 4.2. Mengubah lokasi ke dalam direktori yang berisi file yang telah didownload, mengubah arsip dan menciptakan symlink ke dalam direktori MySQL.

3. Mengubah lokasi ke dalam direktori mysql dan menjalankan script untuk menciptakan *database system*. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.3.

```
shell> cd mysql
shell>. / scripts / mysql install db-user = mysql
```

Kode Sumber 4.3. Mengubah lokasi ke direktori mysql.

4. Mengatur izin yang diperlukan oleh server MSQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.4.

```
shell> chown-R root.
shell> chown-R mysql data
shell> chgrp-R mysql.
```

Kode Sumber 4.4. Mengatur izin yang diperlukan oleh server MSQl.

5. Menyalin *script startup MySQL* ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.5.

```
shell> cp support-files/mysql.server / etc / init.d
/ mysql
shell> chmod + x / etc / init.d / mysql
shell> update-rc.d mysql defaults
```

Kode Sumber 4.5. Menyalin script startup MySQL ke direktori yang sesuai, mengubah menjadi executable, dan memulai ketika sistem beroperasi

4.2.2 Pemasangan Node Management pada node03

Pemasangan node manajemen memerlukan manajemen server MySQL Cluster (ndb_mgmd), diasumsikan bahwa `mysql-cluster-gpl-7.1.5-linux-i686-glibc23.tar.gz` telah ditempatkan di `/var / tmp`. Untuk memasang `ndb_mgmd` dan `ndb_mgm` pada host Cluster, sistem sebagai root melakukan langkah-langkah sebagai berikut:

1. Mengubah lokasi ke dalam direktori `/ var / tmp` direktori, dan mengekstrak `ndb_mgm` dan `ndb_mgmd` dari arsip ke direktori yang sesuai seperti `/ usr / local / bin`. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.6.

```
shell> cd/usr/local
shell> tar-zxvf mysql-cluster-gpl-7.1.34-linux-x86-
glibc23.tar.gz
shell> cd / usr/local/mysql-cluster-gpl-7.1.34-
linux-x86-glibc235
shell> cp bin / ndb_mgm */usr/local/bin
```

Kode Sumber 4.6. Mengubah lokasi ke dalam direktori / var / tmp direktori, mengekstrak ndb_mgm dan ndb_mgmd dari arsip ke direktori yang sesuai seperti / usr / local / bin.

2. Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.7.

```
shell> cd/usr/local/bin
shell> chmod +x ndb mgm*
```

Kode Sumber 4.7. Mengubah lokasi ke dalam direktori tempat file disalin, kemudian dieksekusi.

4.2.3 Konfigurasi Manajemen Node

Konfigurasi pada manajemen node dilakukan dengan langkah-langkah sebagai berikut:

1. Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.8.

```
shell> mkdir / var / lib / mysql-cluster
shell> cd / var / lib / mysql-cluster
vi config.ini
```

Kode Sumber 4.8. Membuat direktori tempat file konfigurasi ditemukan kemudian membuat file itu sendiri.

2. Mengatur file “config.ini”. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.9.


```
[ndbd default]
NoOfReplicas=2
DataMemory=10G
IndexMemory=2G
MaxNoOfAttributes=10000
MaxNumberOfTables=2500
MaxOfOrderedIndexes=4086
MaxOfConcurrentOperations=250000
MaxOfConcurrentOperations=250000
[tcp default]
[ndb_mgmd]
hostname=10.151.64.182
datadir=/var/lib/mysql-cluster
[ndbd]
hostname=10.151.64.182
datadir=/usr/local/mysql/data
[ndbd]
hostname=10.151.64.203
datadir=/usr/local/mysql/data
[mysqld]
MaxNoOfAttributes=10000
hostname=10.151.64.182
[mysqld]
MaxNoOfAttributes=10000
hostname=10.151.64.203
```

Kode Sumber 4.9. Mengatur file “config.ini”.

4.2.4 Konfigurasi Data dan SQL Node

Konfigurasi data dan SQL Node dilakukan dengan cara mengedit file my.cnf pada direktori /etc/. Untuk setiap data node dan SQL node yang diatur pada my.cnf. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.10.

```
[client]
port=3306
socket=/tmp/mysql.sock
[mysqld]
port=3306
socket=/tmp/mysql.sock
ndbcluster
ndb-connectstring=10.151.64.181
[mysql_cluster]
ndb-connectstring=10.151.64.181
```

Kode Sumber 4.10. Data dan SQL Node

4.2.5 Memulai MySQL Cluster

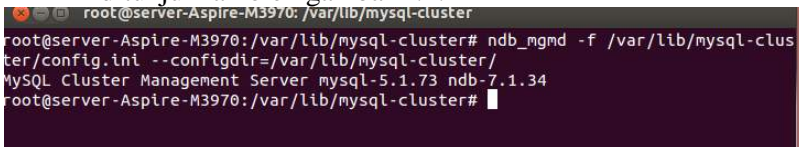
Setiap proses node cluster harus dimulai secara terpisah. Manajemen node harus dimulai terlebih dahulu, kemudian node data. Pada setiap node SQL dilakukan langkah sebagai berikut:

1. Pada node03 (host manajemen), untuk memulai proses manajemen node dari shell sistem dilakukan perintah berikut. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.11.

```
shell> ndb_mgmd -f /var/lib/mysql-cluster/config.ini --configdir=/var/lib/mysql-cluster/
```

Kode Sumber 4.11 Memulai Proses Manajemen Node

Jika berhasil akan muncul tampilan yang ditunjukkan oleh gambar 4.2.



Gambar 4.1 Memulai Proses Manajemen Node

2. Jalankan perintah untuk memulai ndbd dan proses mysql server pada masing-masing Data/host SQL. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.12.

```
shell> /usr/local/mysql/bin/ndbd
```

Kode Sumber 4.12 Memulai Proses ndbd dan proses mysql server

Jika berhasil akan keluar tampilan seperti Gambar 4.2

```

root@master-Aspire-M3970: /home/master
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ^C^Ctrl-C -- ext!
Aborted
root@master-Aspire-M3970:/home/master# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: 10.151.64.181:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0)
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)

[mysqld(API)] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

ndb_mgm>

```

Gambar 4.2 Ndbd dan Proses Mysql Server dapat Dimulai.

3. Mengaktifkan mysql pada data node. Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.11.

```
shell> / etc / init.d / mysql start
```

Kode Sumber 4.13 Memulai proses ndbd dan proses mysql server.

Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.3.

```

root@master-Aspire-M3970: /home/master
root@master-Aspire-M3970:/home/master# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.1.73-ndb-7.1.34-cluster-gpl MySQL Cluster Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| Information_schema |
| db_baru |
| db_baru_1 |
| ezerp |
| mysql |
| ndb_3_fs |
| ndbinfo |
| test |
+-----+
8 rows in set (0.03 sec)

mysql>

```

Gambar 4.3 Mysql pada Data Node Aktif

4. Menguji pada client node dengan perintah `ndb_mgm`. Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.4.



```

root@node2-Aspire-M3970: /home/node2
root@node2-Aspire-M3970:/home/node2# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 182
Server version: 5.1.73-ndb-7.1.34-cluster-gpl MySQL Cluster Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'root'@'10.151.64.182' IDENTIFIED BY '';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'10.151.64.185' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>

```

Gambar 4.4 Data Node Saling Terkoneksi

4.3 Implementasi RBAC (Role Based Access Controll)

Pada bagian ini akan dijelaskan mengenai implementasi *RBAC* yang terdapat pada sistem ERP 2015. Secara rinci mengenai implementasi *RBAC* dijabarkan sebagai berikut.

4.3.1 Membuat Tabel Pengguna

Pembuatan tabel pengguna dilakukan dengan menjalankan perintah `yii migrate` pada folder aplikasi.

Jika berhasil yang maka akan terlihat tampilan yang ditunjukkan oleh Gambar 4.5.

```

Basic usage: composer <command>
For more information just type "composer".

C:\xampp1\htdocs\ezerp>yii migrate
Yii Migration Tool (based on Yii v2.0.4)

Creating migration history table "migration"...done.
Total 1 new migration to be applied:
    m130524_201442_init

Apply the above migration? (yes|no) [no]:yes
*** applying m130524_201442_init
> create table {{%user}} ... done (time: 0.142s)
*** applied m130524_201442_init (time: 0.227s)

Migrated up successfully.

C:\xampp1\htdocs\ezerp>

```

Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil

4.3.2 Membuat 4 Tabel Autentifikasi RBAC dan Tabel Pengguna

Pada tahap ini dibutuhkan 4 tabel autentifikasi yang terdiri dari:

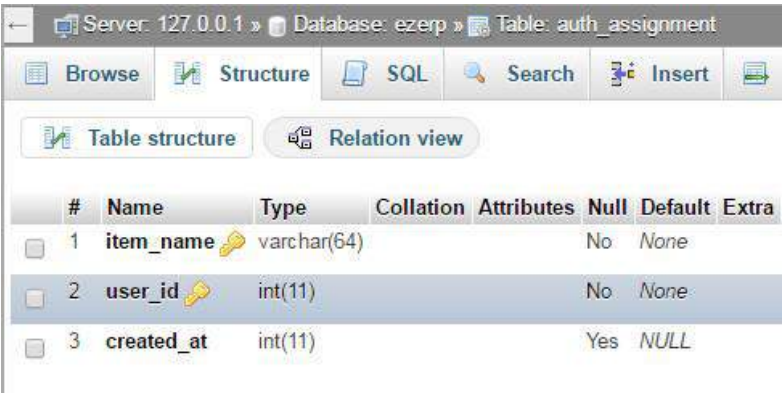
1. Tabel Item
2. Tabel Child
3. Tabel Assignment
4. Tabel Rule

Cara untuk membuat tabel-tabel tersebut yaitu dengan menjalankan perintah yang ditunjukkan pada Kode Sumber 4.6.

```
Yii migrate --migrationPath=@yii/rbac/migrations
```

Kode Sumber 4.14 Generate Tabel RBAC

Jika berhasil yang maka akan terlihat 4 tabel pada phpmyadmin. Tampilan yang ditunjukkan oleh Gambar 4.5, Gambar 4.6, Gambar 4.7, Gambar 4.8

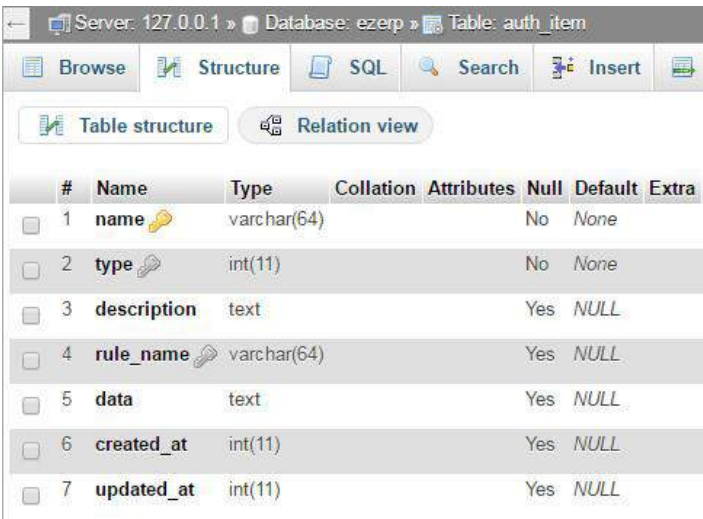


Server: 127.0.0.1 » Database: ezerp » Table: auth_assignment

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	item_name	varchar(64)			No	None	
2	user_id	int(11)			No	None	
3	created_at	int(11)			Yes	NULL	

Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil

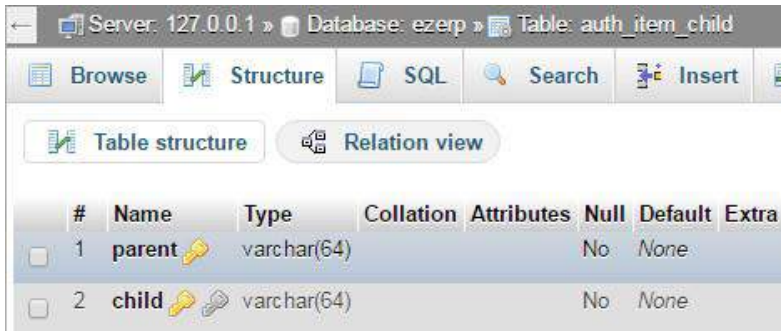


Server: 127.0.0.1 » Database: ezerp » Table: auth_item

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	name	varchar(64)			No	None	
2	type	int(11)			No	None	
3	description	text			Yes	NULL	
4	rule_name	varchar(64)			Yes	NULL	
5	data	text			Yes	NULL	
6	created_at	int(11)			Yes	NULL	
7	updated_at	int(11)			Yes	NULL	

Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil

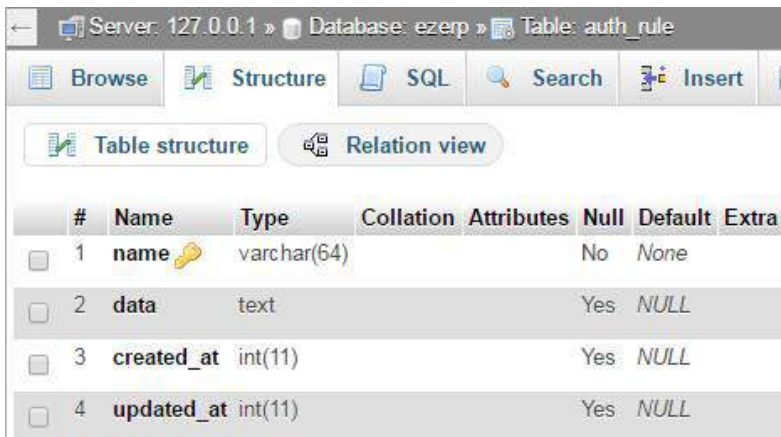


Server: 127.0.0.1 » Database: ezerp » Table: auth_item_child

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	parent	varchar(64)			No	None	
2	child	varchar(64)			No	None	

Gambar 4.5 Proses Migrasi Tabel Pengguna Berhasil



Server: 127.0.0.1 » Database: ezerp » Table: auth_rule

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	name	varchar(64)			No	None	
2	data	text			Yes	NULL	
3	created_at	int(11)			Yes	NULL	
4	updated_at	int(11)			Yes	NULL	

Gambar 4.6. Proses Migrasi Tabel Pengguna Berhasil

4.3.3 Membuat Modul Admin

Proses ini bertujuan untuk meletakkan konfigurasi pengguna dan masing-masing model dari 4 tabel autentifikasi. Kemudian dilakukan konfigurasi pada folder config file web.php. Terdapat 3 konfigurasi yaitu:

1. Admin : digunakan sebagai akses ke modul admin.
2. Auth Manager : sebagai autentifikasi di yii2 dan mengatur role default sebagai *guest*

3. Session Time Out : megatur durasi time out session selama 5 menit atau 300 detik.

Masing –masing konfigurasi tersebut ditunjukkan pada kode sumber.

4.3.4 Membuat Model Tabel Autentifikasi dan Tabel Pengguna di Modul Admin

Pada tahap ini dilakukan *generate* kelas model dari masing-masing tabel autentifikasi pada modul admin yang diperlukan pada tahap sebelumnya. Kemudian ditambahkan *generate* kelas *controller* dan *view* pada tabel pengguna. Proses generate ini menggunakan *yii generator* yang telah disediakan oleh *framework yii2*.

4.3.5 Membuat *Controller* dan *View* tabel pengguna di Modul Admin

4.3.6 Menambahkan Kode pada *UserController* Modul Admin

Pada modul admin, file *usercontroller.php* ditambahkan kode fungsi untuk semua tabel autentifikasi. Masing-masing fungsi ditunjukkan pada Kode Sumber 4.16, Kode Sumber 4.17, Kode Sumber 4.18.

```
public function actionAuthItem()
{
    $auth = Yii::$app->authManager;

    // menambahkan akses sebagai admin ke tabel
    auth_item
        $admin = $auth->createPermission('admin');
        $admin->description = 'Allow user to access
    all page';
        $auth->add($admin);

    // menambahkan akses sebagai asset
    management manajer ke tabel auth_item
        $am_manager = $auth->createPermission('am-
    manager');
```



```

        $am_manager->description = 'Allow user as
Asset Management Manager';
        $auth->add($am_manager);

        // menambahkan akses sebagai asset
management staff ke tabel auth_item
        $am_staff = $auth->createPermission('am-
staff');
        $am_staff->description = 'Allow user as
Asset Management Staff';
        $auth->add($am_staff);

        // menambahkan akses sebagai account
payable manajer ke tabel auth_item
        $ap_manager = $auth->createPermission('ap-
manager');
        $ap_manager->description = 'Allow user as
Account Payable Manager';
        $auth->add($ap_manager);

        // menambahkan akses sebagai account
payable staff ke tabel auth_item
        $ap_staff = $auth->createPermission('ap-
staff');
        $ap_staff->description = 'Allow user as
Account Payable Staff';
        $auth->add($ap_staff);

        // menambahkan akses sebagai account
receivable manajer ke tabel auth_item
        $ar_manager = $auth->createPermission('ar-
manager');
        $ar_manager->description = 'Allow user as
Account Receivable Manager';
        $auth->add($ar_manager);

        // menambahkan akses sebagai account
receivable staff ke tabel auth_item
        $ar_staff = $auth->createPermission('ar-
staff');
        $ar_staff->description = 'Allow user as
Account Receivable Staff';
        $auth->add($ar_staff);

        // menambahkan akses sebagai finance
manajer ke tabel auth_item
        $fi_manager = $auth->createPermission('fi-
manager');

```

```

        $fi_manager->description = 'Allow user as
Finance Manager';
        $auth->add($fi_manager);

        // menambahkan akses sebagai finance staff
        ke tabel auth_item
        $fi_staff = $auth->createPermission('fi-
staff');
        $fi_staff->description = 'Allow user as
Finance Staff';
        $auth->add($fi_staff);

        //      menambahkan      akses      sebagai
accounting/general ledger manajer ke tabel
auth_item
        $gl_manager = $auth->createPermission('gl-
manager');
        $gl_manager->description = 'Allow user as
General Ledger Manager';
        $auth->add($gl_manager);

        //      menambahkan      akses      sebagai
accounting/general ledger staff ke tabel auth_item
        $gl_staff = $auth->createPermission('gl-
staff');
        $gl_staff->description = 'Allow user as
General Ledger Staff';
        $auth->add($gl_staff);

        // menambahkan akses sebagai human resource
management manajer ke tabel auth_item
        $hrm_manager = $auth-
>createPermission('hrm-manager');
        $hrm_manager->description = 'Allow user as
Human and Resource Manager';
        $auth->add($hrm_manager);

        // menambahkan akses sebagai human resource
management staff ke tabel auth_item
        $hrm_staff = $auth->createPermission('hrm-
staff');
        $hrm_staff->description = 'Allow user as
Human and Resource Staff';
        $auth->add($hrm_staff);

        // menambahkan akses sebagai inventory and
warehouse management manajer ke tabel auth_item
        $iwm_manager = $auth-
>createPermission('iwm-manager');

```

```

        $iwm_manager->description = 'Allow user as
Inventory and Warehouse Management Manager';
        $auth->add($iwm_manager);

        // menambahkan akses sebagai inventory and
warehouse management staff ke tabel auth_item
        $iwm_staff = $auth->createPermission('iwm-
staff');
        $iwm_staff->description = 'Allow user as
Inventory and Warehouse Management Staff';
        $auth->add($iwm_staff);

        // menambahkan akses sebagai production
planning manajer ke tabel auth_item
        $pp_manager = $auth->createPermission('pp-
manager');
        $pp_manager->description = 'Allow user as
Producton Planning Manager';
        $auth->add($pp_manager);

        // menambahkan akses sebagai production
planning staff ke tabel auth_item
        $pp_staff = $auth->createPermission('pp-
staff');
        $pp_staff->description = 'Allow user as
Producton Planning Staff';
        $auth->add($pp_staff);

        // menambahkan akses sebagai purchasing
manajer ke tabel auth_item
        $pur_manager = $auth-
>createPermission('pur-manager');
        $pur_manager->description = 'Allow user as
Purchasing Manager';
        $auth->add($pur_manager);

        // menambahkan akses sebagai purchasing
staff ke tabel auth_item
        $pur_staff = $auth->createPermission('pur-
staff');
        $pur_staff->description = 'Allow user as
Purchasing Staff';
        $auth->add($pur_staff);

        // menambahkan akses sebagai sales and
distribution manajer ke tabel auth_item
        $sd_manager = $auth->createPermission('sd-
manager');

```

```

        $sd_manager->description = 'Allow user as
        Sales and Distribution Manager';
        $auth->add($sd_manager);

        // menambahkan akses sebagai sales and
        distribution staff ke tabel auth_item
        $sd_staff = $auth->createPermission('sd-
        staff');
        $sd_staff->description = 'Allow user as
        Sales and Distribution Staff';
        $auth->add($sd_staff);
    }

```

Kode Sumber 4.15 Kode Fungsi Tabel AuthItem

```

public function actionItemChild(){
    $auth = Yii::$app->authManager;

    //admin dapat mengakses semua daftar izin akses
    $am_manager = $auth->createPermission('am-
    manager');
    $am_staff = $auth->createPermission('am-staff');
    $ap_manager = $auth->createPermission('ap-
    manager');
    $ap_staff = $auth->createPermission('ap-staff');
    $ar_manager = $auth->createPermission('ar-
    manager');
    $ar_staff = $auth->createPermission('ar-staff');
    $fi_manager = $auth->createPermission('fi-
    manager');
    $fi_staff = $auth->createPermission('fi-staff');
    $gl_manager = $auth->createPermission('gl-
    manager');
    $gl_staff = $auth->createPermission('gl-staff');
    $hrm_manager = $auth->createPermission('hrm-
    manager');
    $hrm_staff = $auth->createPermission('hrm-staff');
    $iwm_manager = $auth->createPermission('iwm-
    manager');
    $iwm_staff = $auth->createPermission('iwm-staff');
    $pp_manager = $auth->createPermission('pp-
    manager');
    $pp_staff = $auth->createPermission('pp-staff');
    $pur_manager = $auth->createPermission('pur-
    manager');
    $pur_staff = $auth->createPermission('pur-staff');
    $sd_manager = $auth->createPermission('sd-
    manager');
    $sd_staff = $auth->createPermission('sd-staff');
}

```

```

$admin = $auth->createRole('admin');
$auth->add($admin);
$auth->addChild($admin, $am_manager);
$auth->addChild($admin, $am_staff);
.....
}

```

Kode Sumber 4.16 Kode Fungsi Tabel ItemChild

```

public function actionAuthAssignment(){
    $auth = Yii::$app->authManager;

    $admin = $auth->createRole('admin');

    $auth->assign($admin, 1);
}

```

Kode Sumber 4.17 Kode Fungsi Tabel AuthAssignment

4.4 Implementasi *Multitenancy*

Pada subbab ini akan dibahas mengenai implementasi alur proses aplikasi yang telah dirancang pada Bab III. Alur proses aplikasi akan dibahas mulai dari proses implementasi multitenan.

4.4.1 Menambahkan Database Untuk Tenant Baru

Pada tahap ini dilakukan penambahan database untuk tenant baru dengan cara melakukan konfigurasi pada server node mysql. Kode implementasi yang dimaksud ditunjukkan oleh Kode Sumber 4.19.

```

`mysql -u root -p;` lalu tekan enter
`create database <nama database>;` lalu tekan enter
`use <nama database>;` lalu tekan enter
e /tmp/mysql-dump/final-db.sql

```

Kode Sumber 4.18 Penambahan Database Tenant Baru

4.4.2 Login Tenant

Setelah proses pembuatan database dan replikasi pada tahap 4.4.2 selesai, maka tenant melakukan login dengan memilih salah satu database, nama database yang dipilih tersebut disimpan dalam session dan akan digunakan untuk koneksi yang akan dibuat.

Kode implementasi yang dimaksud ditunjukkan oleh kode sumber 4.20.

```

$host = '10.151.64.182'; // node mysql
$port = 3306;
$waitTimeoutInSeconds = 1;
$src =
    @fsockopen($host,$port,$errCode,$errStr,$waitTime
outInSeconds);
if(is_resource($src)){
    $_SESSION['dbserver_ip'] = "10.151.64.182"; // node
mysql
    // penyimpanan nama database sebagai session untuk
multitenancy
    $dbname = isset($_SESSION['database_name']) ?
    $_SESSION['database_name'] : 'test';
} else {
    $_SESSION['dbserver_ip'] = "10.151.64.203"; // node
mysql
}

$conection = [
    'class' => 'yii\db\Connection',
    'dsn' =>
        'mysql:host='.$_SESSION['dbserver_ip'].';dbname='
        .$dbname.'',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8'
];
return $conection;

```

Kode Sumber 4.19 Login Tenant

4.5 Implementasi Modul *Human Resource Management*

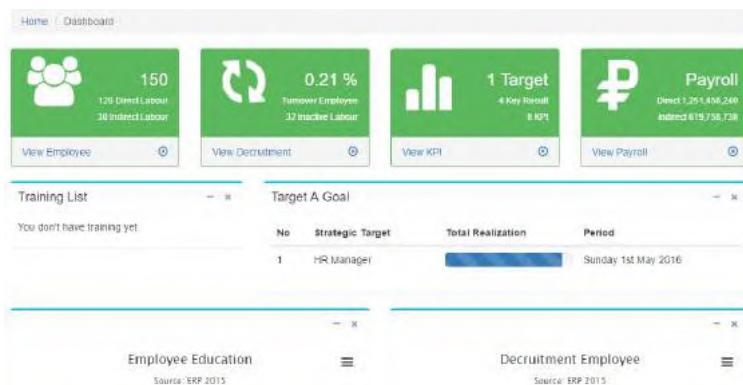
4.5.1 Implementasi Antarmuka

Pada subbab ini dibahas implementasi antarmuka berdasarkan rancangan antarmuka yang dibahas pada bab sebelumnya. Berikut penjelasan mengenai antarmuka pada aplikasi ERP khususnya modul HRM.

4.5.1.1 Antarmuka Halaman *Dashboard*

Halaman *Dashboard* modul HRM yang dibangun ditunjukkan oleh Gambar 4.7. Dalam halaman *dashboard*,

pengguna dapat melihat informasi data sejumlah data yang ada di modul HRM. Pengguna juga dapat memonitor informasi secara *real-time*. Terdapat 4 kotak utama pada bagian atas antarmuka ini yakni, informasi jumlah pegawai, *turnover rate* pegawai, Jumlah KPI, dan total gaji. Kemudian pada bagian bawah terdapat grafik jumlah pegawai berdasarkan pendidikan terakhir yang dimiliki pegawai dan grafik pengurangan pegawai berdasarkan alasan pengurangan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.7.



Gambar 4.7 Antarmuka Halaman *Dashboard*

Home / HRM Job Requisition

HRM Job Requisition

Create New

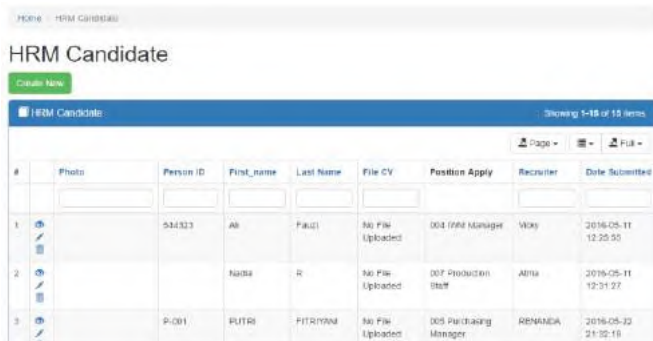
HRM Job Requisition Showing 1-16 of 16 items.

#	Hrm Position	Job Requisition Code	Job Requisition Name	Requirement
1	007 PS	JOBREQ0001	Production Staff	System analyst
2	006 PUM	JOBREQ0002	Purchasing Manager	Purchasing Manager
3	001 EM	JOBREQ0004	Executive Manager	Executive Manager
4	001 ES	JOBREQ0005	Executive Staff	Executive Staff
5	002 FAM	JOBREQ0006	Financial Accounting Manager	Financial Accounting Manager
6	003 FAR	JOBREQ0007	Financial Accounting Staff	Financial Accounting Staff

Gambar 4.8 Antarmuka Halaman Job Requisition

4.5.1.2 Antarmuka Halaman *Candidate*

Pada antarmuka ini pengguna dapat melihat daftar data *Candidate* yang telah dibuat. Pengguna juga dapat memilih kegiatan menyunting, melihat detail data, dan menghapus data *Candidate* tertentu serta membuat data *Candidate* yang baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.9.



HRM Candidate

Create New

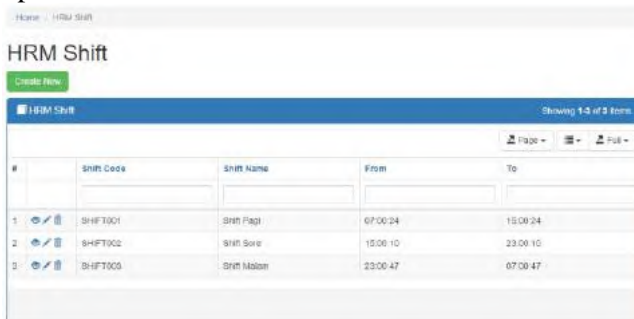
Showing 1-3 of 3 items

#	Photo	Person ID	First_name	Last Name	File CV	Position Apply	Recruiter	Date Submitted
1		SM323	Ab	Pauli	No File Uploaded	004 HRH Manager	Wahy	2016-05-11 12:35:00
2			Nadisa	R	No File Uploaded	007 Production Staff	Adma	2016-05-11 12:31:27
3		P-001	ELITE	PITRIPRANA	No File Uploaded	003 Purchasing Manager	REHANEWA	2016-05-30 21:52:16

Gambar 4.9 Antarmuka Halaman *Candidate*

4.5.1.3 Antarmuka Halaman *Shift*

Pada antarmuka ini pengguna dapat melihat daftar data *Shift* yang telah dibuat. Pengguna juga dapat memilih kegiatan menyunting, melihat detail data, dan menghapus data *Shift* tertentu serta membuat data *Shift* yang baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.10.



HRM Shift

Create New

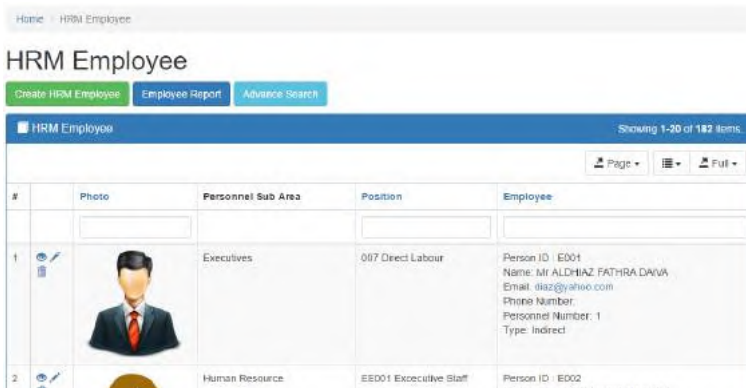
Showing 1-3 of 3 items

#	Shift Code	Shift Name	From	To
1	SHIFT001	Shift Pagi	07:00:24	15:00:24
2	SHIFT002	Shift Sore	15:00:10	23:00:10
3	SHIFT003	Shift Malam	23:00:47	07:00:47

Gambar 4.10 Antarmuka Halaman *Shift*

4.5.1.4 Antarmuka Halaman *Employee*

Pada antarmuka ini pengguna dapat melihat daftar data *Employee* yang telah dibuat. Pengguna juga dapat memilih kegiatan menyunting, melihat detail data, dan menghapus data *Employee* tertentu serta membuat data *Employee* yang baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.11.



Gambar 4.11 Antarmuka Halaman *Employee*

4.5.1.5 Antarmuka Halaman *Key Performance Indicator*

Pada antarmuka ini pengguna dapat melihat daftar data *Key Performance Indicator* yang telah dibuat. Pengguna juga dapat memilih kegiatan menyunting, melihat detail data, dan menghapus data *Key Performance Indicator* tertentu serta membuat data *Key Performance Indicator* yang baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.12.

Home / Hrm Key Performance Indicator

Hrm Key Performance Indicator

Create Hrm Key Performance Indicator Advance Search

Hrm Key Performance Indicator Showing 1-8 of 8 Items

Page - - Full -

#	Hrm Key Result Area	Kpi Detail	Weight	Target	Realisasi	Skor	Final Skor	Type
1	Recruitment	% jumlah rekrutasi pegawai baru yang dapat dipenuhi dengan tepat waktu (< 15 hari)	15	100	90	90	14	max
2	Recruitment	Rata-rata skor evaluasi karyawan baru setelah 3 bulan masa percobaan	15	80	82	100	15	max
3	Training and Development	Jumlah jam training per karyawan (per kapita) per tahun	10	30	28	93	5	max

Gambar 4.12 Antarmuka Halaman *Key Performance Indicator*

4.5.1.6 Antarmuka Halaman *Attendance List*

Pada antarmuka ini pengguna dapat melihat daftar data *Attendance List* yang telah dibuat. Pengguna juga dapat memilih kegiatan menyunting, melihat detail data, dan menghapus data *Attendance List* tertentu serta membuat data *Attendance List* yang baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.13.

Home / HRM Attendance List

HRM Attendance List

Create HRM Attendance List Generate Attendance

HRM Attendance List Showing 1-20 of 13942 Items

Page - - Full -

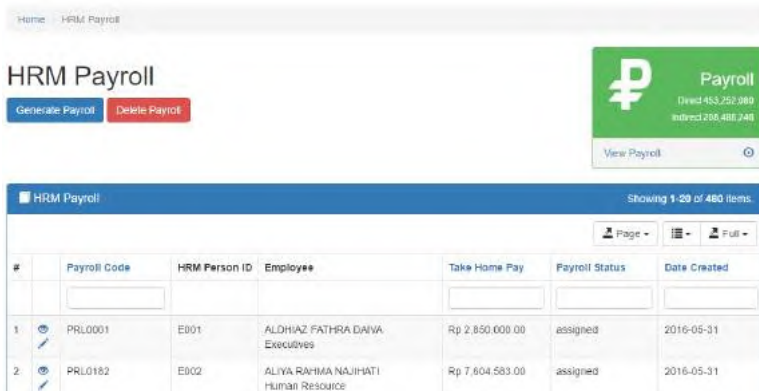
#	Hrm Employee	Attendance Name	Attendance Date	Start Work	End Work	Duration	Overtime
1	ED01	N	2016-05-02	07:00:20	16:00:20	9	0
2	ED01	N	2016-05-03	07:00:20	16:00:20	9	0
3	ED01	N	2016-05-04	07:00:20	16:00:20	9	0
4	ED01	N	2016-05-05	07:00:20	16:00:20	9	0

Gambar 4.13 Antarmuka Halaman *Attendance List*

4.5.1.7 Antarmuka Halaman *Payroll*

Pada antarmuka ini pengguna dapat melihat daftar data *Payroll* yang telah dibuat. Pengguna juga dapat memilih kegiatan

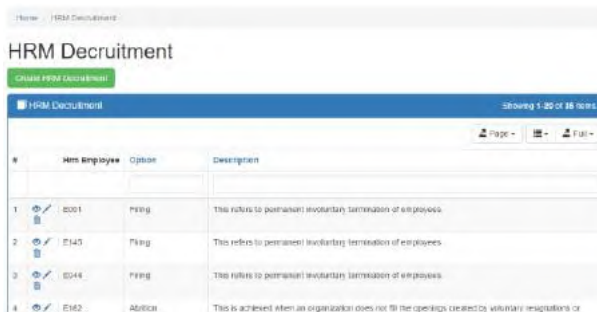
menyunting, melihat detail data, dan menghapus data *Payroll* tertentu serta membuat data *Payroll* yang baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.14.



Gambar 4.14 Antarmuka Halaman *Payroll*

4.5.1.8 Antarmuka Halaman *Decruitment*

Pada antarmuka ini pengguna dapat melihat daftar data *Decruitment* yang telah dibuat. Pengguna juga dapat memilih kegiatan menyunting, melihat detail data, dan menghapus data *Decruitment* tertentu serta membuat data *Decruitment* yang baru. Tampilan antarmuka ini dapat dilihat pada Gambar 4.15. Halaman antarmuka ini juga digunakan pada setiap data subdomain level 2. Halaman antarmuka ini meliputi halaman *index*, *form create and update*, dan *view*.



Gambar 4.15 Antarmuka Halaman *Decruitment*

4.5.2 Implementasi Program

Implementasi program berisikan tentang kode-kode program yang terdapat pada Tugas Akhir ini. Subbab ini terbagi dalam beberapa bagian.

4.5.2.1 Program Halaman *Dashboard*

Halaman utama modul *Human Resource Management* menampilkan informasi dari data-data yang ada di modul *HRM*. Pengaturan tampilan tersebut diatur di kelas *DashboardController* dalam fungsi *actionIndex()* . Kode tampilan yang dimaksud ditunjukkan oleh kode sumber 4.20.

1	public function actionIndex(){
2	\$day = "2016-04-30";
3	\$profit = 0.0;
4	\$lastProfit = 0;
5	\$models = IstatementController::actionCari(0, \$day, 3, 0);
6	foreach (\$models as \$key => \$value) {
7	if(\$value['acc_name'] == "Revenues" \$value['acc_name'] == "Revenue") {
8	\$profit += \$value['balance'];
9	\$lastProfit += \$value['balanceOld'];
10	}
11	if(\$value['acc_name'] == "Expenses" \$value['acc_name'] == "Expense") {
12	\$profit -= \$value['balance'];
13	\$lastProfit -= \$value['balanceOld'];
14	}
15	}
16	\$data_employee = new HrmEmployee;
17	\$id = 1;
18	\$getDirectLabour = HrmEmployee::find()->where("type = 'Direct' and status = 'Active'")->count();

19	\$getIndirectLabour = HrmEmployee::find()->where("type = 'Indirect' and status = 'Active')->count();
20	\$now = HrmEmployee::find()->where("status = 'Active')->count();
21	\$out = HrmEmployee::find()->where("status = 'Inactive')->count();
22	\$training = HrmTraining::find()->where("time > NOW()")->orderBy(['time' => SORT_DESC])->asArray()->all();
23	\$getAllStrategicTarget = HrmStrategicTarget::find()->orderBy(['period' => SORT_DESC])->asArray()->all();
24	\$getStrategicTarget = HrmStrategicTarget::find()->count();
25	\$getResultArea = HrmKeyResultArea::find()->asArray()->count();
26	\$getKeyPerformanceIndicator= HrmKeyPerformanceIndicator::find()->count();
27	\$db = Yii::\$app->db;
28	\$DirectSalary = \$db->createCommand(\$getSumGajiDirect)->queryOne();
29	\$IndirectSalary = \$db->createCommand(\$getSumGajiIndirect)->queryOne();
30	\$kpii = \$db->createCommand(\$kpi)->queryAll();
31	\$biday = \$db->createCommand(\$bday)->queryAll();
32	\$male = \$db->createCommand(\$querymale)->queryOne();
33	\$female = \$db->createCommand(\$queryfemale)->queryOne();
34	return \$this->render('index',[
35	'profit'=>\$profit,
36	'getDirectLabour' => \$getDirectLabour,
37	'getIndirectLabour' => \$getIndirectLabour,
38	'now' => \$now,
39	'out' => \$out,
40	'training' => \$training,
41	'getStrategicTarget' => \$getStrategicTarget,
42	'getResultArea' => \$getResultArea,
43	'getKeyPerformanceIndicator' => \$getKeyPerformanceIndicator,
44	'getAllStrategicTarget' => \$getAllStrategicTarget,
45	'directSalary' => \$DirectSalary['take_home_pay'],

46	'indirectSalary' => \$IndirectSalary['take_home_pay'],
47	'kpi' => \$kpi,
48	'bday' => \$bday,
49	'male' => \$male,
50	'female' => \$female,
51]);
52	}

Kode Sumber 4.20 Kode Sumber Halaman *Dashboard*

4.5.2.2 Program Melihat *Job Requisition*

Pada implementasi melihat daftar *Job Requisition*, sistem menampilkan seluruh daftar *Job Requisition* yang berhasil ditambahkan pada *Job Requisition*. Pengaturan tampilan ini diatur pada kelas *HrmJobRequisitionController* fungsi *actionIndex()* Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.21.

1	public function actionIndex()
2	{
3	\$searchModel = new HrmJobRequisitionSearch();
4	\$dataProvider = \$searchModel->
5	search(Yii::\$app->request->queryParams);
6	return \$this->render('index', [
7	'searchModel' => \$searchModel,
8	'dataProvider' => \$dataProvider,
9]);
10	}

Kode Sumber 4.21 Kode Sumber Melihat *Job Requisition*

Pada potongan kode program ini juga digunakan pada submodul-submodul seperti: *Candidate*, *Hire Candidate*, *Skill*, *Level*, *Shift*, *Training*, *KPI*, *Master Payroll*, *Payroll* dan *Decruitment*.

4.5.2.3 Program Menambah *Job Requisition*

Pada implementasi menambah daftar *Job Requisition*, sistem menambahkan data *Job Requisition*. Pengaturan manambah

transaksi baru ini diatur pada kelas *HrmJobRequisitionController* fungsi *actionCreate()*. Kode untuk menampilkan seluruh transaksi tersebut ditunjukkan pada kode sumber 4.23.

```

1 public function actionCreate()
2     {
3         $model = new HrmJobRequisition();
4
5         if ($model->loadAll(Yii::$app->request-
6 >post())) {
7             if (!$model->job_requisition_code) {
8                 $old_code = HrmJobRequisition::find()-
9 >orderBy(['id'=>SORT_DESC])->limit(1)->one();
10
11                 if ($old_code == null) {
12                     $sc = "JOBREQ0001";
13                 } else if ($old_code->id < 9) {
14                     $new = $old_code->id + 1;
15                     $sc = "JOBREQ000".$new;
16                 } else if ($old_code->id < 99) {
17                     $new = $old_code->id + 1;
18                     $sc = "JOBREQ00".$new;
19                 } else if ($old_code->id < 999) {
20                     $new = $old_code->id + 1;
21                     $sc = "JOBREQ0".$new;
22                 }
23                 $model->job_requisition_code = $sc;
24             }
25             $model->saveAll();
26             return $this->redirect(['view', 'id' =>
27 $model->id]);
28         } else {
29             return $this->renderAjax('create', [
30                 'model' => $model,
31             ]);
32         }
33     }

```

Kode Sumber 4.22 Kode Sumber Menambah *Job Requisition*

Pada potongan kode program ini juga digunakan pada submodul-submodul seperti: *Candidate*, *Hire Candidate*, *Skill*, *Level*, *Shift*, *Training*, *KPI*, *Master Payroll*, *Payroll* dan *Decruitment*.

4.5.2.4 Program Menyunting *Job Requisition*

Pada implementasi menyunting *Job Requisition*, sistem memperbarui data transaksi pada *Job Requisition*. Pengaturan menyunting transaksi ini diatur pada kelas *HrmJobRequisitionController* fungsi *actionUpdate()*. Kode untuk menyunting *Job Requisition* ditunjukkan pada kode sumber 4.24.

1	public function actionUpdate(\$id)
2	{
3	\$model = \$this->findModel(\$id);
4	
5	if (\$model->loadAll(Yii::\$app->request->post()) && \$model->saveAll()) {
6	return \$this->redirect(['view', 'id' => \$model->id]);
7	} else {
8	return \$this->render('update', [
9	'model' => \$model,
10]);
11	}
12	}

Kode Sumber 4.23 Kode Sumber Menyunting *Job Requisition*

4.5.2.5 Program Menghapus *Job Requisition*

Pada implementasi menghapus *Job Requisition*, sistem menghapus data transaksi pada *Job Requisition*. Pengaturan menyunting transaksi ini diatur pada kelas *HrmJobRequisitionController* fungsi *actionDelete()*. Kode untuk menghapus *Job Requisition* ditunjukkan pada kode sumber 4.25.

1	public function actionDelete(\$id){
2	\$this->findModel(\$id)-
3	>deleteWithRelated();
4	return \$this->redirect(['index']);
5	}

Kode Sumber 4.24 Kode Sumber Menghapus *Job Requisition*

Pada potongan kode program ini juga digunakan pada submodul-submodul seperti: *Candidate*, *Hire Candidate*, *Skill*, *Level*, *Shift*, *Training*, *KPI*, *Master Payroll*, *Payroll* dan *Decruitment*.

4.5.2.6 Program Rejected Candidate

Pada implementasi *Rejected Candidate*, sistem memperbarui status pada data *Candidate*. Pengaturan program ini diatur pada kelas *HrmHireCandidateController* fungsi *actionRemove()*. Kode untuk menyunting *Hire Candidate* ditunjukkan pada kode sumber 4.25.

1	public function actionRemove(\$id){
2	\$model = \$this->findModel(\$id);
3	\$model->decision = "2";
4	\$model->update();
5	return \$this->redirect(['index']);
6	}

Kode Sumber 4.25 Kode Sumber Rejected Candidate

4.5.2.7 Program Menyunting Shift

Pada implementasi menyunting *Shift*, sistem memperbarui data transaksi pada *Shift*. Pengaturan menyunting transaksi ini diatur pada kelas *HrmShiftController* fungsi *actionUpdate()*. Kode untuk menyunting *Shift* ditunjukkan pada kode sumber 4.26.

1	public function actionUpdate(\$id){
2	\$model = \$this->findModel(\$id);
3	
4	if (\$model->loadAll(Yii::\$app->request->post())) {
5	\$f = date("H:i:s",strtotime(\$model->from));
6	\$t = date("H:i:s",strtotime(\$model->to));
7	\$model->from = \$f;
8	\$model->to = \$t;
9	
10	\$queryEmployeeId = "select id
11	from hrm_employee
12	where status =
13	'Active' and type = 'Direct' and id
14	NOT IN (select
15	employee_id from hrm_hasshift)";
16	\$db = Yii::\$app->db;
17	\$employeeId = \$db->createCommand(\$queryEmployeeId)->queryAll();
18	\$jumlahEmployee = sizeof(\$employeeId);

```

18         if( $jumlahEmployee <= (int)$model-
19 >numberOfLabour){
19             for($i=0;$i<sizeof($employeeId);$i++){
20                 $insertEmployee = "insert into
hrm_hasshift (employee_id,shift_id) values
20 ('".$employeeId[$i]['id'].",'".$model->id."'");
                $db-
21 >createCommand($insertEmployee)->execute();
22             }
23         }else {
24             for($i=0;$i<(int)$model-
24 >numberOfLabour;$i++){
                $insertEmployee = "insert into
hrm_hasshift (employee_id,shift_id) values
25 ('".$employeeId[$i]['id'].",'".$model->id."'");
                $db-
26 >createCommand($insertEmployee)->execute();
27             }
28         }
29
30         $hasshift = new HrmHasshift();
31         $model->saveAll();
32         return $this->redirect(['view', 'id' =>
$model->id]);
33     } else {
34         return $this->render('update', [
35             'model' => $model,
36         ]);
37     }
38 }

```

Kode Sumber 4.26 Kode Sumber Menyunting Shift

4.5.2.8 Program Generate Attendance List

Pada implementasi membuat daftar hadir dalam satu bulan. Sistem akan menampilkan form untuk meminta nilai bulan dan jam masuk untuk pegawai 'Indirect'. Perngaturan tampilan ini diatur pada kelas *HrmAttendanceListController* fungsi *actionGenerate()*. Kode untuk mengenerate daftar hadir ditunjukkan pada kode sumber 4.27.

```

1 public function actionGenerate(){
2     $model = new HrmAttendanceList();
3     if ($model->loadAll(Yii::$app->request-
>post())) {
4         $queryEmployeeId = "select id,type from
hrm_employee where status = 'Active'";
5         $db = Yii::$app->db;

```


	if(\$shift['from'] >
35	\$shift['to'])){
	\$dura = '24:00:00' -
36	\$shift['from'] + \$shift['to'];
37	}else{
	\$dura = \$shift['to'] -
38	\$shift['from'];
39	}
40	
	\$queryInsert = "insert into
	hrm_attendance_list
	(employee_id,attendance_name,attendance_date,start_wor
	k,end_work,duration,created_at,updated_at) values
	(".\$employeeId[\$i]['id'].",'N',STR_TO_DATE('".\$\$tin."'
	,'%Y-%m-
	%d'),'".\$\$shift['from'].",'".\$\$shift['to'].",'".\$\$dura."
41	,NOW(),NOW())";
	\$db-
42	>createCommand(\$queryInsert)->execute();
43	}
44	
45	}
46	}
47	
48	return \$this->redirect(['index']);
49	} else {
50	return \$this->redirect(['index']);
51	}
52	}

Kode Sumber 4.27 Kode Sumber Generate Attendance List.

4.5.2.9 Program Generate Payroll

Pada implementasi menghitung gaji setiap pegawai dalam satu bulan. Sistem akan menampilkan form untuk meminta nilai bulan dan tanggal akhir bulan. Pengaturan tampilan ini diatur pada kelas *HrmPayrollController* fungsi *actionGenerate()*. Kode untuk mengenerate daftar hadir ditunjukkan pada kode sumber 4.28.

1	public function actionGenerate(){
2	\$model = new HrmPayroll();
	if(\$model->loadAll(Yii::\$app->request-
3	>post())){
	\$queryEmployeeId = "select
	id,type,marital_status from hrm_employee where
4	status = 'Active'";
5	\$db = Yii::\$app->db;

```

        $employeeId = $db-
6 >createCommand($queryEmployeeId)->queryAll();
7 $to = $model->payroll_date;
    $from = date("Y-m-d", strtotime("-
8 ".$model->day." days", strtotime($to)));
9 // var_dump($from);
10 // exit();
11 for($i=0;$i<sizeof($employeeId);$i++){
12
        $queryOvertime = "select
sum(overtime) as overtime from hrm_attendance_list
where employee_id = '". $employeeId[$i]['id']."' and
attendance_date >= ' ".$from."' and attendance_date
13 <= ' ".$to."'";
        $queryWorkhour = "select
sum(duration) as workhour from hrm_attendance_list
where employee_id = '". $employeeId[$i]['id']."' and
attendance_date >= ' ".$from."' and attendance_date
14 <= ' ".$to."'";
        $queryNormal = "select
count(attendance_name) as normal from
hrm_attendance_list where attendance_name = 'N' and
employee_id = '". $employeeId[$i]['id']."' and
attendance_date >= ' ".$from."' and attendance_date
15 <= ' ".$to."'";
        $queryVacation = "select
count(attendance_name) as vacation from
hrm_attendance_list where attendance_name = 'V' and
employee_id = '". $employeeId[$i]['id']."' and
attendance_date >= ' ".$from."' and attendance_date
16 <= ' ".$to."'";
        $queryAbsense = "select
count(attendance_name) as absense from
hrm_attendance_list where attendance_name = 'A' and
employee_id = '". $employeeId[$i]['id']."' and
attendance_date >= ' ".$from."' and attendance_date
17 <= ' ".$to."'";
        $querySick = "select
count(attendance_name) as sick from
hrm_attendance_list where attendance name = 'S' and
employee_id = '". $employeeId[$i]['id']."' and
attendance_date >= ' ".$from."' and attendance_date
18 <= ' ".$to."'";
        $queryBasicSalary = "select
basic_salary from hrm_position where id = (select
position_id from hrm_hasemployee where employee_id =
19 '". $employeeId[$i]['id']."'");
        $queryPositionAllowance = "select
20 position allowance as pa from hrm position where id

```

```

= (select position_id from hrm_hasemployee where
employee_id = '".EmployeeId[$i]['id'].'"");
$queryMasterPayroll = "select * from
21 hrm_master_payroll";
22
        $overtime = $db-
23 >createCommand($queryOvertime)->queryAll();
        $workhour = $db-
24 >createCommand($queryWorkhour)->queryAll();
        $normal = $db-
25 >createCommand($queryNormal)->queryAll();
        $vacation = $db-
26 >createCommand($queryVacation)->queryAll();
        $absense = $db-
27 >createCommand($queryAbsense)->queryOne();
        $sick = $db-
28 >createCommand($querySick)->queryAll();
        $basicSalary = $db-
29 >createCommand($queryBasicSalary)->queryOne();
        $positionAllowance = $db-
30 >createCommand($queryPositionAllowance)->queryOne();
        $mp = $db-
31 >createCommand($queryMasterPayroll)->queryAll();
32 $total_deduction = 0;
33 $total_income = 0;
34 $take_home_pay = 0;
35
        //generate kode
36 $hasil = "";
37 if($model->payroll_code == null){
38     $sufix = HrmPayroll::find()-
39 >orderBy(['id'=>SORT_DESC])->limit(1)->one();
40     if($sufix == null){
41         $a = "";
42     } else {
43         $a = $sufix->id;
44     }
45     $prefix = "PRL";
46     $hasil = $this-
47 >generateCode($prefix,$a);
48     }
49
        // deduction antara direct dan
50 indirect jika tidak masuk
51 $abs = 0;
52 $abs2 = 0;
53 if($employeeId[$i]['type'] ==
'Direct'){
        //hitung deduction

```

54	if(\$normal[0]['normal'] != 0){
	\$abs =
	((int)\$absense['absense']*(int)\$basicSalary['basic_s
55	alary'])/(int)\$normal[0]['normal'];
56	}
	\$total_deduction =
57	\$total_deduction + \$abs;
58	}else{
59	// hitung deduction
60	if(\$normal[0]['normal'] != 0){
	\$abs2 =
	(int)\$absense['absense']*(int)\$basicSalary['basic_sa
61	lary']/(int)\$normal[0]['normal'];
62	}
63	
64	\$total_deduction =
	\$total_deduction + \$abs2;
65	}
66	
67	
68	// update gaji bruto
	\$total_income = \$total_income +
	(int)\$basicSalary['basic_salary'] +
69	(int)\$positionAllowance['pa'];
70	
	//menghitung lembur berdasarkan
71	aturan Kepmenakertrans No. 102/MEN/VI/2004
72	\$totalLembur = 0;
	if(\$overtime[0]['overtime'] !=
73	NULL){
	\$a =
74	(int)\$overtime[0]['overtime'];
75	
76	for(\$u=0;\$u<\$a;\$u++){
77	if(\$u=0){
	\$totalLembur =
	\$totalLembur + 1.5 * \$a * 1/173 *
	((int)\$basicSalary['basic_salary'] +
78	(int)\$positionAllowance['pa']);
79	}else{
	\$totalLembur =
	\$totalLembur + 2 * \$a * 1/173 *
	((int)\$basicSalary['basic_salary'] +
80	(int)\$positionAllowance['pa']);
81	}
82	}
83	
	\$total_income = \$total_income +
84	\$totalLembur;

```

85         }
86
87
88         //hitung dari master payroll
        $queryFromMaster = "select * from
89 hrm_master_payroll";
        $master = $db-
90 >createCommand($queryFromMaster)->queryAll();
91
92         if($master != null){
93
94         for($iter=0;$iter<sizeof($master);$iter++){
95             if($master[$iter]['type'] ==
96             "Deduction"){
97
98             if((int)$master[$iter]['amount'] <=100){
99                 $total_deduction =
100                 $total_deduction +
101                 (int)$master[$iter]['amount']*(int)$basicSalary['bas
102                 ic_salary']/100;
103             }else{
104                 $total_deduction =
105                 $total_deduction + (int)$master[$iter]['amount'];
106             }
107         }else{
108
109         if((int)$master[$iter]['amount'] <=100){
110             $total_income =
111             $total_income +
112             (int)$master[$iter]['amount']*(int)$basicSalary['bas
113             ic_salary']/100;
114         }else{
115             $total_income =
116             $total_income + (int)$master[$iter]['amount'];
117         }
118     }
119 }
120
121 //hitung Pajak Penghasilan Pph 21
122 Tahun 2015
123 /*
124
125 */
126 $getTanggungan = "select password
127 from hrm_employee where id =
128 ".$employeeId[$i]['id'].>";

```



```

117         $jumlahtang = $db-
118 >createCommand($getTanggungan)->queryOne();
119         $netto = 0;
120         $pph = 0;
121         $bruto = $total_income * 12;
122         if($employeeId[$i]['marital_status']
123 == 'Single'){
124             if ($jumlahtang['password'] ==
125 '0') {
126                 $netto = $bruto - 54000000 -
127 $total_deduction;
128                 if($netto < 0){
129                     $netto = 0;
130                 }
131             }else if($jumlahtang['password']
132 == '1'){
133                 $netto = $bruto - 58500000 -
134 $total_deduction;
135                 if($netto < 0){
136                     $netto = 0;
137                 }
138             }else if($jumlahtang['password']
139 == '2'){
140                 $netto = $bruto - 63000000 -
141 $total_deduction;
142                 if($netto < 0){
143                     $netto = 0;
144                 }
145             }else {
146                 $netto = $bruto - 67500000 -
147 $total_deduction;
148                 if($netto < 0){
149                     $netto = 0;
150                 }
151             }
152         }else{
153             if ($jumlahtang['password'] ==
154 '0') {
155                 $netto = $bruto - 58500000 -
156 $total_deduction;
157                 if($netto < 0){
158                     $netto = 0;
159                 }
160             }else if($jumlahtang['password']
161 == '1'){

```

```

153     $netto = $bruto - 63000000 -
154     $total_deduction;
155         if($netto < 0){
156             $netto = 0;
157         }
158     }else if($jumlahtang['password']
159 == '2'){
160         $netto = $bruto - 67500000 -
161         $total_deduction;
162         if($netto < 0){
163             $netto = 0;
164         }
165     }else {
166         $netto = $bruto - 72000000 -
167         $total_deduction;
168         if($netto < 0){
169             $netto = 0;
170         }
171     }
172     if($netto <= 50000000){
173         $pph = 0.05*$netto/12;
174     }else if($netto <= 250000000){
175         $pph = 0.15*$netto/12;
176     }else if($netto <= 500000000){
177         $pph = 0.25*$netto/12;
178     }else {
179         $pph = 0.3*$netto/12;
180     }
181     $total_deduction = $total_deduction
182 + $pph;
183 //hitung akhri take_home_pay
184 $take_home_pay = $take_home_pay +
185 $total_income - $total_deduction - $pph;
186 if($workhour[0]['workhour'] == 0){
187     $take_home_pay = 0;
188     $total_income = 0;
189     $total_deduction = 0;
190 }
191 // perhitungan total gaji
192 $query = "insert into hrm_payroll
    (payroll_code,employee_id,total_income,total_deducti
    on,take_home_pay,created_at,updated_at) values
    ('".$hasil."','".$employeeId[$i]['id']."','".$total_i

```

```

ncome.", ".$total_deduction.", ".$take_home_pay.", ".$
to.", NOW())");
193
194         $db = Yii::$app->db;
195
196         $db->createCommand($query)-
197 >execute();
198
199         $test = HrmPayroll::find()-
200 >orderBy(['id'=>SORT_DESC])->limit(1)->one();
201
202         $insertpph = "insert into
203 hrm_deduction
204 (payroll_id,deduction_name,deduction_amount,created_
205 at,updated_at) values ( ".$test-
206 >id.", 'PPH21', ".$pph.", ".$sto.", NOW())";
207         $db->createCommand($insertpph)-
208 >execute();
209         // echo "<pre>";
210         // var_dump($test->id);
211         // echo "</pre>";
212         // exit();
213         if($employeeId[$i]['type'] ==
214 'Direct'){
215         $queryInsertDeduction1 = "insert
216 into hrm_deduction
217 (payroll_id,deduction_name,deduction_amount,created_
218 at,updated_at) values ( ".$test-
219 >id.", 'Absence', ".$abs.", ".$sto.", NOW())";
220         $db-
221 >createCommand($queryInsertDeduction1)->execute();
222         }else{
223         $queryInsertDeduction = "insert
224 into hrm_deduction
225 (payroll_id,deduction_name,deduction_amount,created_
226 at,updated_at) values ( ".$test-
227 >id.", 'Absence', ".$abs2.", ".$sto.", NOW())";
228         $db-
229 >createCommand($queryInsertDeduction)->execute();
230         }
231
232         // insert ke tabel hrm_income
233         $insertGajiPokok = "insert into
234 hrm_income
235 (payroll_id,income_name,amount,created_at,updated_at
236 ) values ( ".$test->id.", 'Basic
237 Salary', (int)$basicSalary['basic_salary'], ".$sto
238 .", NOW())";

```

	<pre> \$insertTunjanganTetap = "insert into hrm_income (payload_id,income_name,amount,created_at,updated_at) values (". \$test->id.", 'Position Allowance',". (int)\$positionAllowance['pa']"." ,"". \$to. 217 "' ,NOW()) "; \$db- 218 >createCommand(\$insertGajiPokok)->execute(); \$db- 219 >createCommand(\$insertTunjanganTetap)->execute(); 220 221 //insert potongan dan tunjangan 222 if(\$master != null){ 223 for(\$iter=0;\$iter<sizeof(\$master);\$iter++){ 224 if(\$master[\$iter]['type'] == "Deduction"){ 225 if((int)\$master[\$iter]['amount'] <=100){ \$master[\$iter]['amount'] = (int)\$master[\$iter]['amount']*(int)\$basicSalary['bas 226 ic_salary']/100; \$queryInsertDeduction2 = "insert into hrm_deduction (payload_id,deduction_name,deduction_amount,created_ at,updated_at) values (". \$test- >id.","". \$master[\$iter]['name']"." ,"". \$master[\$iter][' 227 'amount']"." ,"". \$to."',NOW()) "; \$db- 228 >createCommand(\$queryInsertDeduction2)->execute(); 229 }else{ \$queryInsertDeduction2 = "insert into hrm_deduction (payload_id,deduction_name,deduction_amount,created_ at,updated_at) values (". \$test- >id.","". \$master[\$iter]['name']"." ,"". (int)\$master[\$i 230 ter]['amount']"." ,"". \$to."',NOW()) "; \$db- 231 >createCommand(\$queryInsertDeduction2)->execute(); 232 } 233 }else{ 234 235 if((int)\$master[\$iter]['amount'] <=100){ 260 }</pre>
--	---

Kode Sumber 4.28 Kode Sumber *Generate Payroll*

4.5.2.10 Program Menghapus *Payroll*

Pada implementasi menghapus *Payroll*, sistem menghapus data transaksi pada *Payroll*. Pengaturan menyunting transaksi ini diatur pada kelas *HrmPayrollController* fungsi *actionDelete()*. Kode untuk menghapus *Payroll* ditunjukkan pada kode sumber 4.29.

```

1 public function actionDelete(){
2     $model = new HrmPayroll();
3     if($model->loadAll(Yii::$app->request-
4 >post())){
5         $db = Yii::$app->db;
6         $queryDelete = "delete from hrm_payroll
7 where created_at = '". $model->payroll_date."'";
8         $db->createCommand($queryDelete)-
9 >execute();
10         return $this->redirect(['index']);
11     }else {
12         return $this->redirect(['index']);
13     }
14 }

```

Kode Sumber 4.29 Kode Sumber Menghapus *Payroll*

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan adalah pengujian terhadap kebutuhan fungsionalitas sistem yang telah dijabarkan pada Bab III dan terhadap tujuan dibuatnya aplikasi ini, yakni agar partisipan tertarik untuk melakukan terapi guna mempercepat proses penyembuhan gejala vertigo yang dialaminya.

5.1 Lingkungan Pengujian

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan sistem pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Basis data yang digunakan pada *server* adalah *MySQL Cluster*.
2. 2 PC untuk *server* basis data menggunakan Sistem Operasi Ubuntu 12.04.
3. PC untuk *server* menggunakan Intel® Core™ i3-2120 @3.30GHz , RAM 4GB dengan Sistem Operasi Windows 8.1 Enterprise x64.
4. Mozilla Firefox 46.0.1 dan Chrome 49.0 sebagai antarmuka untuk pengujian aplikasi klien.

5.2 Business Plan

ERP merupakan aplikasi skala besar yang digunakan oleh perusahaan dalam menjalankan *business plan* yang telah dibuat secara matang dengan perhitungan matematis. Pada Tugas Akhir ini telah dibuat sebuah *business plan* yang menjadi acuan dalam menjalankan aplikasi ERP secara keseluruhan.

Business plan yang dibuat adalah sebuah perusahaan manufaktur sepeda. Perusahaan manufaktur dipilih karena memiliki tingkat kompleksitas proses bisnis yang paling tinggi.

Sehingga aplikasi ERP yang dikembangkan dapat mengakomodasi proses bisnis berbagai jenis perusahaan.

- **Daftar Barang Jadi**

Tabel 5.1. Daftar Barang Jadi

No	Nama Barang
1	Red Deluxe Touring Bike
2	Red Profesional Touring Bike

Pada Tabel 5.1, ditunjukkan bahwa perusahaan ini memproduksi dua jenis sepeda. Dua jenis sepeda tersebut memiliki bahan dasar yang berbeda, Deluxe Touring Bike berbahan dasar aluminum, sedangkan Profesional Touring Bike berbahan dasar karbon.

- **Daftar Aset**

Tabel 5.2. Daftar Aset

No.	Nama	Kuantitas
1	Tanah	180 x 150 m ²
2	Kantor	50 x 50 m ²
3	Parkir	60 x 50 m ²
4	Kantin	40 x 60 m ²
5	Pabrik	108 x 80 m ²
6	Pengolahan Limbah	20 x 40 m ²
7	Raw Materials Inventory	40 x 60 m ²
8	Semi-Finished Good Inventory	40 x 60 m ²
9	Finished Good Inventory	40 x 60 m ²
10	Welding Machine	2 baris
11	Molding Machine	2 baris
12	Laser Cutting Machine	2 baris
13	Spray Painting Machine	2 baris
14	Testing Machine	2 baris
15	Truck	10 buah
16	Forklift	6 buah

17	Heavy Forklift	6 buah
-----------	----------------	--------

Pada Tabel 5.2, perusahaan memiliki 17 asset pada perencanaannya dengan kuantitas dari masing-masing aset yang ada.

- **Daftar Bahan Baku**

Tabel 5.3 Daftar Bahan Baku

No.	Nama bahan baku	Kuantitas	Satuan unit
1	Tire	1.2	Pcs
2	Seat Kit	0.5	Pcs
3	Chain	1.5	Pcs
4	Gear	1.8	Pcs
5	Brake	0.6	Pcs
6	Handle Bar	1.2	Pcs
7	Pedal	0.4	Pcs
8	Aluminium	2	m
9	Carbon Fiber	2	m
10	Red Paint 20KG	20	Big Drum
11	Velg	2.3	Pcs
12	Tube	0.4	Pcs
13	Hex Nut 5mm	0.04	Pcs
14	Lock Washer 5mm	0.06	Pcs
15	Socket Head Bolt 5mm	0.03	Pcs

Perencanaan yang diadakan oleh perusahaan ini akan menggunakan 15 macam bahan baku seperti yang dijabarkan pada Tabel 5.3. 15 bahan baku ini ditentukan dari daftar material yang akan diterangkan di Tabel 3.4.

Daftar material yang dijabarkan pada Tabel 5.4. menjelaskan material-material yang dibutuhkan untuk memproduksi barang jadi yang dibutuhkan pada perusahaan ini.

- **Daftar Material**

Tabel 5.4. Daftar material

No.	Bahan yang dibutuhkan	Kuantitas	Satuan unit	Bahan yang dihasilkan
1	Aluminium	5	m	Kerangka untuk Red Deluxe Touring Bike
2	Paint drum with Red Color 20KG	0.05	Big drum	
3	Carbon Fiber	5	m	
4	Paint drum with Red Color 20KG	0.05	Big drum	Kerangka untuk Red Profesional Touring Bike
5	Hex Nut 5 mm	2	Pcs	
6	Lock Washer 5 mm	2	Pcs	
7	Socket Head Bolt 5mm	1	Pcs	Roda sepeda
8	Tire's Bike	1	Pcs	
9	Wheel's Tube	1	Pcs	
10	Wheel's Velg	1	Pcs	
11	Frame for Deluxe Touring Bike with Red Color	1	Pcs	Red Deluxe Touring Bike
12	Wheel's Bike	2	Pcs	
13	Chain's Bike	1	Pcs	
14	Handle Bar's Bike	1	Pcs	
15	Seat Kit's Bike	1	Pcs	
16	Pedal's Bike	2	Pcs	
17	Gear's Bike	1	Pcs	
18	Front and Rear Brake's Bike	1	Pcs	

No.	Bahan yang dibutuhkan	Kuantitas	Satuan unit	Bahan yang dihasilkan
19	Frame for Deluxe Touring Bike with Red Color	1	Pcs	Red Professional Touring Bike
20	Wheel's Bike	2	Pcs	
21	Chain's Bike	1	Pcs	
22	Handle Bar's Bike	1	Pcs	
23	Seat Kit's Bike	1	Pcs	
24	Pedal's Bike	2	Pcs	
25	Gear's Bike	1	Pcs	
26	Front and Rear Brake's Bike	1	Pcs	

• Sumberdaya

Pada perusahaan ini memiliki beberapa sumber daya yang terdiri atas beberapa pegawai dan beberapa mesin yang terpisah menjadi 2 (dua) baris. Terdapat 150 orang sebagai *direct labour*, 32 orang sebagai *indirect labour*. Untuk mesin, terdapat *welding machine*, *molding machine*, *laser cutting machine*, *spray machine*, dan *testing machine* yang masing-masing mesin untuk 2 (dua) baris.

• Perencanaan keuangan

Perencanaan keuangan dalam perusahaan ini digunakan untuk merincikan keuangan yang digunakan untuk pengelolaan keuangan dalam perusahaan. Perencanaan keuangan ditunjukkan pada Tabel 5.5.

Tabel 5.5. Perencanaan keuangan perusahaan

No.	Nama perencanaan keuangan	Jumlah (dalam Rp)
1	Total pembelian aset aktif	27.997.311.200,00
2	Total pembelian bahan baku	88.615.744.062,50
3	Biaya lain-lain	3,386,944,737.50
4	Peminjaman dari Bank	120.000.000.000,00

5.3 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian dilakukan dalam tiga tahap yaitu pengujian kebutuhan fungsionalitas, pengujian kegunaan sistem, dan pengujian perbandingan *metrics*. Pengujian kebutuhan fungsionalitas menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada hasil keluaran sistem.

5.3.1 Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas aplikasi dilakukan secara mandiri dengan melakukan skenario yang sama dengan rancangan alur proses aplikasi sebagai tolok ukur keberhasilan pengujian, dan mengacu pada kasus penggunaan yang sebelumnya telah dijelaskan pada Bab III. Pengujian pada kebutuhan fungsionalitas dapat dijabarkan pada subbab berikut:

5.3.1.1 Pengujian Mengelola Struktur Perusahaan

Pengujian ini terdiri dari pengujian mengelola data struktur perusahaan. Pengujian mengelola data struktur perusahaan yaitu menambah, menyunting, dan menghapus data struktur perusahaan yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.6.

Tabel 5.6 Pengujian Mengelola Struktur Perusahaan

ID	UJ.UC-001
Referensi	UCTA-001
Nama	Mengelola struktur perusahaan
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data struktur perusahaan
Skenario 1	Pengguna menambah data struktur perusahaan
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Org Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna masuk ke halaman <i>Client, Company, Personnel Area, Personnel Sub</i>

	Area dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail struktur perusahaan
Skenario 2	Pengguna menyunting data struktur perusahaan
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Org Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data struktur perusahaan yang telah ada, kemudian menyunting data tersebut pada halaman <i>Company</i> , <i>Personnel Area</i> , <i>Personnel Sub Area</i> dan menyimpan data yang baru ke dalam basis data
Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail rute pengiriman
Skenario 3	Pengguna menghapus data struktur perusahaan
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Org Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data struktur perusahaan yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data

Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama struktur perusahaan

5.3.1.2 Pengujian Mengelola Struktur Kepegawaian

Pengujian ini terdiri dari pengujian mengelola data struktur kepegawaian. Pengujian mengelola data struktur kepegawaian yaitu menambah, menyunting, dan menghapus data struktur kepegawaian yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.7.

Tabel 5.7 Pengujian Mengelola Struktur Kepegawaian

ID	UJ.UC-002
Referensi	UCTA-002
Nama	Mengelola struktur kepegawaian
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data struktur kepegawaian
Skenario 1	Pengguna menambah data struktur kepegawaian
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Employee Structure</i>
Data Uji	Data struktur kepegawaian
Langkah Pengujian	Pengguna masuk ke halaman <i>Job, Position</i> dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail struktur perusahaan
Skenario 2	Pengguna menyunting data struktur kepegawaian
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Employee Structure</i>
Data Uji	Data struktur perusahaan

Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data struktur perusahaan yang telah ada, kemudian menyunting data tersebut pada halaman <i>Job, Position</i> dan menyimpan data yang baru ke dalam basis data
Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail rute pengiriman
Skenario 3	Pengguna menghapus data struktur kepegawaian
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Employee Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data struktur perusahaan yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama struktur perusahaan

5.3.1.3 Pengujian Mengelola Perekrutan Pegawai

Pengujian ini terdiri dari pengujian mengelola data perekrutan pegawai. Pengujian mengelola data struktur perusahaan yaitu menambah, menyunting, dan menghapus data perekrutan pegawai yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.8.

Tabel 5.8 Pengujian Mengelola Perekrutan Pegawai

ID	UJ.UC-003
Referensi	UCTA-003
Nama	Mengelola perekrutan pegawai
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data perekrutan pegawai.
Skenario 1	Pengguna menambah data perekrutan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Recruitment and Hiring</i> .
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna masuk ke halaman <i>Job Requisition, Candidate, Hire Candidate</i> dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail.
Skenario 2	Pengguna menyunting data perekrutan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Recruitment and Hiring</i>
Data Uji	Data perekrutan pegawai
Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data struktur perusahaan yang telah ada, kemudian menyunting data tersebut pada halaman <i>Job Requisition, Candidate, Hire Candidate</i> dan menyimpan data yang baru ke dalam basis data
Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail
Skenario 3	Pengguna menghapus data perekrutan pegawai

Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Recruitment and Hiring</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data struktur perusahaan yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama perekrutan pegawai.

5.3.1.4 Pengujian Mengelola Penempatan Pegawai

Pengujian ini terdiri dari pengujian mengelola penempatan pegawai. Pengujian mengelola data penempatan pegawai yaitu menambah, menyunting, dan menghapus data penempatan pegawai yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.9.

Tabel 5.9 Pengujian Mengelola Penempatan Pegawai

ID	UJ.UC-004
Referensi	UCTA-004
Nama	Mengelola penempatan pegawai
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data penempatan pegawai.
Skenario 1	Pengguna menambah data penempatan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Assignment</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna masuk ke halaman <i>Skill, Level, Shift, Employee</i> dan memasukkan data uji ke dalam basis data

Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail.
Skenario 2	Pengguna menyunting data penempatan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Assignment</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data struktur perusahaan yang telah ada, kemudian menyunting data tersebut pada halaman <i>Skill, Level, Shift, Employee</i> dan menyimpan data yang baru ke dalam basis data
Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail.
Skenario 3	Pengguna menghapus data penempatan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Assignment</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data struktur perusahaan yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama penempatan pegawai

5.3.1.5 Pengujian Mengelola Training

Pengujian ini terdiri dari pengujian mengelola data training. Pengujian mengelola data training yaitu menambah, menyunting, dan menghapus data training yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.10.

Tabel 5.10 Pengujian Mengelola Training

ID	UJ.UC-005
Referensi	UCTA-005
Nama	Mengelola training
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data training
Skenario 1	Pengguna menambah data training
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Org Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna masuk ke halaman <i>Training</i> dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail training
Skenario 2	Pengguna menyunting data struktur perusahaan
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Org Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data struktur perusahaan yang telah ada, kemudian menyunting data tersebut pada halaman <i>Client, Company, Personnel Area, Personnel Sub Area</i> dan menyimpan data yang baru ke dalam basis data

Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail rute pengiriman
Skenario 3	Pengguna menghapus data training
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Org Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data training yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama training

5.3.1.6 Pengujian Mengelola Bisnis Goal

Pengujian ini terdiri dari pengujian mengelola data bisnis goal. Pengujian mengelola data struktur perusahaan yaitu menambah, menyunting, dan menghapus data struktur perusahaan yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.11.

Tabel 5.11 Pengujian Mengelola Bisnis Goal

ID	UJ.UC-006
Referensi	UCTA-006
Nama	Mengelola struktur perusahaan
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data struktur perusahaan
Skenario 1	Pengguna menambah data bisnis goal
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Target A Goal</i>

Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna masuk ke halaman <i>Strategic Traget</i> , <i>Key Result Area</i> , <i>Key Performance Indicator</i> dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail <i>Strategic Target</i>
Skenario 2	Pengguna menyunting data bisnis goal
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Target A Goal</i>
Data Uji	Data <i>Target A Goal</i>
Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data struktur perusahaan yang telah ada, kemudian menyunting data tersebut pada halaman <i>Strategic Traget</i> , <i>Key Result Area</i> , <i>Key Performance Indicator</i> dan menyimpan data yang baru ke dalam basis data
Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail.
Skenario 3	Pengguna menghapus data bisnis goal
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Target A Goal</i>
Data Uji	Data bisnis goal
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data bisnis yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data

Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama struktur perusahaan

5.3.1.7 Pengujian Mengelola Penggajian Pegawai

Pengujian ini terdiri dari pengujian mengelola data struktur perusahaan. Pengujian mengelola data struktur perusahaan yaitu menambah, menyunting, dan menghapus data struktur perusahaan yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.12.

Tabel 5.12 Pengujian Mengelola Penggajian Pegawai

ID	UJ.UC-007
Referensi	UCTA-007
Nama	Mengelola penggajian pegawai
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data Penggajian Pegawai
Skenario 1	Pengguna menambah data penggajian
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Payroll</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna masuk ke halaman <i>Attendance List, Billing Rate, Master Payroll, Payroll</i> dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail struktur perusahaan
Skenario 2	Pengguna menyunting data struktur perusahaan
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Payroll</i>
Data Uji	Data struktur perusahaan

Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data penggajian pegawai yang telah ada, kemudian menyunting data tersebut pada halaman <i>Attendance List</i> , <i>Billing Rate</i> , <i>Master Payroll</i> , <i>Payroll</i> dan menyimpan data yang baru ke dalam basis data
Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail rute pengiriman
Skenario 3	Pengguna menghapus data penggajian pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Payroll</i>
Data Uji	Data penggajian pegawai
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data Penggajian Pegawai yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama Penggajian Pegawai

5.3.1.8 Pengujian Mengelola Pengurangan Pegawai

Pengujian ini terdiri dari pengujian mengelola data pengurangan pegawai. Pengujian mengelola data pengurangan pegawai yaitu menambah, menyunting, dan menghapus data pengurangan pegawai yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.13.

Tabel 5.13 Pengujian Mengelola Pengurangan Pegawai

ID	UJ.UC-008
Referensi	UCTA-008
Nama	Mengelola pengurangan pegawai
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola data pengurangan pegawai.
Skenario 1	Pengguna menambah data pengurangan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Decruitment</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna masuk ke halaman <i>Decruitment</i> dan memasukkan data uji ke dalam basis data
Hasil Yang Diharapkan	Data yang dimasukkan tersimpan di basis data
Hasil Yang Didapat	Data tersimpan di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail struktur perusahaan
Skenario 2	Pengguna menyunting data pengurangan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Decruitment</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>update</i> dari salah satu data struktur perusahaan yang telah ada, kemudian menyunting data tersebut pada halaman <i>Decruitment</i> dan menyimpan data yang baru ke dalam basis data
Hasil Yang Diharapkan	Data baru yang dimasukkan, menggantikan data lama yang ada di dalam basis data
Hasil Yang Didapat	Data baru tersimpan di dalam basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail rute pengiriman

Skenario 3	Pengguna menghapus data struktur perusahaan
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Org Structure</i>
Data Uji	Data struktur perusahaan
Langkah Pengujian	Pengguna memilih tombol <i>delete</i> dari salah satu data struktur perusahaan yang telah ada, kemudian melakukan konfirmasi penghapusan data dan data akan dihapus dari basis data
Hasil Yang Diharapkan	Data yang dihapus akan terhapus dari basis data
Hasil Yang Didapat	Data terhapus di basis data
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama Pengurangan Pegawai

5.3.1.9 Pengujian Melihat Halaman Dashboard

Pengujian ini terdiri dari pengujian mengelola data struktur perusahaan. Pengujian mengelola data struktur perusahaan yaitu menambah, menyunting, dan menghapus data struktur perusahaan yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.114.

Tabel 5.14 Pengujian Melihat Halaman Dashboard

ID	UJ.UC-009
Referensi	UCTA-009
Nama	Melihat halaman dashboard
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat halaman dashboard
Skenario 1	Pengguna melihat halaman dashboard
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu dashboard
Data Uji	Data <i>pegawai, struktur, target a goal</i>
Langkah Pengujian	Pengguna masuk ke halaman dashboard
Hasil Yang Diharapkan	Data ditampilkan di halaman dashboard

Hasil Yang Didapat	Data ditampilkan di halaman dashboard
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman dashboard

5.3.1.10 Pengujian Melihat Laporan Pegawai

Pengujian ini terdiri dari pengujian melihat laporan pegawai. Pengujian melihat laporan pegawai yaitu melihat laporan pegawai yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.15.

Tabel 5.15 Pengujian Melihat Laporan Pegawai

ID	UJ.UC-010
Referensi	UCTA-010
Nama	Melihat laporan Pegawai
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat laporan pegawai
Skenario 1	Pengguna melihat laporan pegawai
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Employee Report</i>
Data Uji	Data <i>employee</i>
Langkah Pengujian	Pengguna masuk ke halaman laporan pegawai.
Hasil Yang Diharapkan	Data ditampilkan di halaman laporan pegawai.
Hasil Yang Didapat	Data tertampilkan di halaman laporan pegawai.
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman laporan pegawai

5.3.1.11 Pengujian Melihat Data Training

Pengujian ini terdiri dari pengujian melihat data training. Pengujian melihat data training yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.16.

Tabel 5.16 Pengujian Melihat Data Training

ID	UJ.UC-011
Referensi	UCTA-011
Nama	Melihat data training
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data <i>training</i>
Skenario 1	Pengguna menambah data <i>training</i>
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>training</i>
Data Uji	Data <i>Training</i>
Langkah Pengujian	Pengguna masuk ke halaman detail <i>training</i> dari data <i>training</i> yang dipilih
Hasil Yang Diharapkan	Data ditampilkan di halaman detail <i>training</i>
Hasil Yang Didapat	Data tertampilkan di halaman detail <i>training</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail <i>training</i>

5.3.1.12 Pengujian Melihat Data Pegawai

Pengujian ini terdiri dari pengujian melihat data pegawai. Pengujian melihat data pegawai yang sudah ada pada sistem. Rincian skenario pengujian pada kasus penggunaan ini dapat dilihat pada Tabel 5.17.

Tabel 5.17 Pengujian Melihat Data Pegawai

ID	UJ.UC-012
Referensi	UC-012
Nama	Melihat detail pegawai
Tujuan Pengujian	Menguji kemampuan sistem dalam melihat data <i>pegawai</i>
Skenario 1	Pengguna menambah data <i>pegawai</i>
Kondisi Awal	Pengguna berada pada menu utama dan memilih menu <i>Employee</i>
Data Uji	Data <i>employee</i>

Langkah Pengujian	Pengguna masuk ke halaman detail <i>employee</i> dari data <i>employee</i> yang dipilih
Hasil Yang Diharapkan	Data ditampilkan di halaman detail <i>employee</i>
Hasil Yang Didapat	Data tertampilkan di halaman detail <i>employee</i>
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman detail <i>employee</i>

5.3.2 Pengujian Fitur Basis Data Terdistribusi

Pengujian Basis Data Terdistribusi (BDT) dilakukan untuk menguji 2 aspek utama dalam konsep BDT, yaitu replikasi dan *high availability*.

Sistem BDT menggunakan 4 server fisik. Dengan rincian sebagai berikut:

1. *Server aplikasi*, adalah server yang di dalamnya terdapat *web server* beserta file aplikasi ERP.
2. *Management Node*, adalah *server* yang berfungsi sebagai pusat pengaturan sistem BDT.
3. *Data Node*, adalah 2 buah server penyimpan data.

5.3.2.1 Replikasi

1. Memastikan seluruh *server* dan sistem BDT dalam keadaan *online*. Pada Gambar 5.1, ditunjukkan *tenant* yang sedang *online*.

Warehouse 3	
ID	3
Company	EZTENANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT

Gambar 5.1. Pengujian fitur replikasi pada sistem

2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi ERP yang ditunjukkan pada Gambar 5.2.

```

master@master-Aspire-M397Q: ~
mysql> use test
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from twm_warehouse where 1;
+-----+
| id | company_id | ws_name          | created_at      | updated_at      |
+-----+
| 3 | 1 | EZERP Testing BDT | NULL            | NULL            |
| 1 | 1 | Maju              | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL            | NULL            |
+-----+
3 rows in set (0.01 sec)

mysql>

```

Gambar 5.2. Pengujian fitur replikasi pada database server1

3. Melakukan pengecekan terhadap hasil penambahan, perubahan, dan penghapusan data pada seluruh data *node* yang ditampilkan pada Gambar 5.3.

```

Database changed
mysql> select * from twm_warehouse where 1;
+-----+
| id | company_id | ws_name          | created_at      | updated_at      |
+-----+
| 3 | 1 | EZERP Testing BDT | NULL            | NULL            |
| 1 | 1 | Maju              | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL            | NULL            |
+-----+

```

Gambar 5.3. Pengujian fitur replikasi pada database server 2

5.3.2.2 *High Availability*

Aspek lain dari BDT adalah *high availability* adalah kemampuan sistem basis data untuk tetap berjalan normal pada kondisi sebuah *node data* tidak aktif. Operasi basis data yang terjadi akan dieksekusi pada *node data* yang aktif. Di lain sisi,

management node akan menciptakan *log* operasi-operasi yang terjadi. Sehingga operasi-operasi basis data dapat dieksekusi pada *node data* yang kembali aktif. Berikut langkah-langkah pengujian aspek *high availability* :

1. Mematikan *server* aplikasi, *management server* dan sistem BDT pada salah satu *node*. Ditampilkan pada Gambar 5.4.

Warehouse 3	
ID	3
Company	EZTENANT BIKE INDONESIA
Warehouse Name	EZERP Testing BDT HA

Gambar 5.4. Pengujian Fitur *High-Availability* pada Sistem

2. Melakukan penambahan, perubahan, dan penghapusan data melalui aplikasi yang ditunjukkan pada Gambar 5.7 dan Gambar 5.5.

```
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0. *)
[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)
[mysqld(API)] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)
ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect form 10.151.64.128)
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0. *)
[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)
[mysqld(API)] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)
```

Gambar 5.5. Pengecekan Availability pada Node Server

```
mysql> use test
Reading table information for completion of table and column names
you can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from iwm_warehouse where 1;
```

id	company_id	ws_name	created_at	updated_at
3	1	EZERP Testing BDT	NULL	NULL
1	1	Maju	2016-03-22 21:08:14	2016-03-22 21:08:14
2	1	EZERP BIKE INDONESIA 2	NULL	NULL

```
3 rows in set (0.01 sec)
```

Gambar 5.6. Pengecekan Database pada Node 1

3. Melakukan pengecekan terhadap hasil pemrosesan data pada *node* yang masih aktif yang ditampilkan pada Gambar 5.9.

```
root@node2-Aspire-M1970: /home/node2
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

[ndb_mgnd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)

[mysqld(API)] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

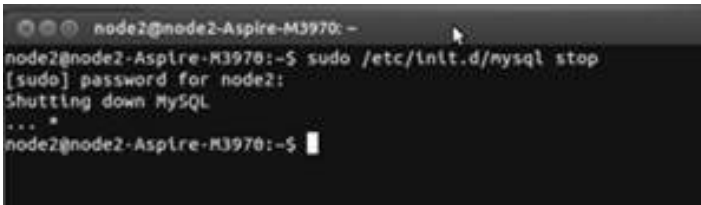
ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect from 10.151.64.182)
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

[ndb_mgnd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)

[mysqld(API)] 2 node(s)
id=4 (not connected, accepting connect from 10.151.64.182)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

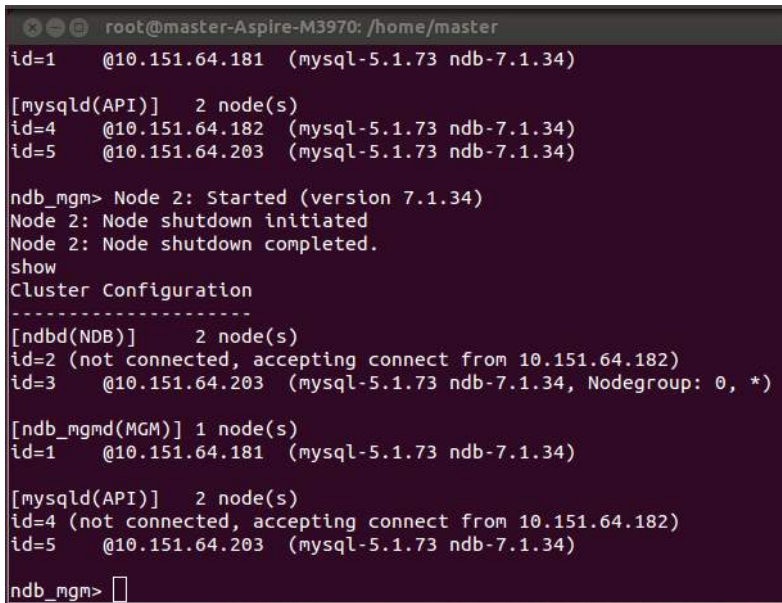
ndb_mgm>
```

Gambar 5.7. Pengecekan availability pada node server



```
node2@node2-Aspire-M3970: -
node2@node2-Aspire-M3970:~$ sudo /etc/init.d/mysql stop
[sudo] password for node2:
Shutting down MySQL
...
node2@node2-Aspire-M3970:~$
```

Gambar 5.8 Mematikan Database Server pada node 2



```
root@master-Aspire-M3970: /home/master
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)

[mysqld(API)] 2 node(s)
id=4 @10.151.64.182 (mysql-5.1.73 ndb-7.1.34)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

ndb_mgm> Node 2: Started (version 7.1.34)
Node 2: Node shutdown initiated
Node 2: Node shutdown completed.
show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect from 10.151.64.182)
id=3 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34, Nodegroup: 0, *)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @10.151.64.181 (mysql-5.1.73 ndb-7.1.34)

[mysqld(API)] 2 node(s)
id=4 (not connected, accepting connect from 10.151.64.182)
id=5 @10.151.64.203 (mysql-5.1.73 ndb-7.1.34)

ndb_mgm>
```

Gambar 5.9 Proses *High Availability* (Sinkronisasi)

Pada Gambar 5.9 merupakan proses dari sinkronisasi data ketika node yang sebelumnya mati dan kembali dinyalakan. Data-data yang tidak ada di dalam node yang mati akan dikopikan data dari data node yang lain.


```
master@master-Aspire-M3970: ~
| tax
| uom
| user
+-----+
228 rows in set (0.01 sec)

mysql> select * from iwm_warehouse where 1;
+-----+-----+-----+-----+-----+
| id | company_id | ws_name | created_at | updated_at |
+-----+-----+-----+-----+-----+
| 3 | 1 | EZERP Testing BDT HA | NULL | NULL |
| 1 | 1 | Maju | 2016-03-22 21:08:14 | 2016-03-22 21:08:14 |
| 2 | 1 | EZERP BIKE INDONESIA 2 | NULL | NULL |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Gambar 5.10 Hasil dari Fitur *High Availability*

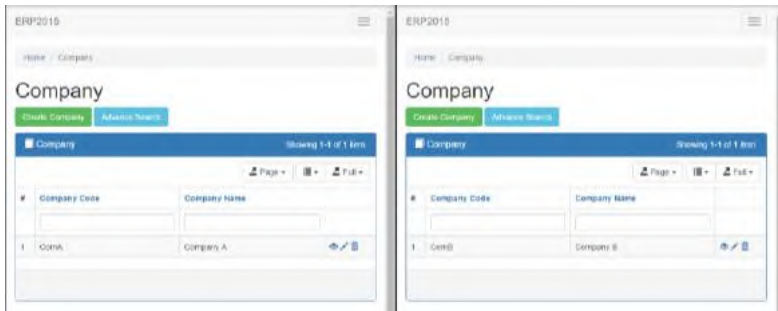
5.3.3 Pengujian Fitur Multitenancy

Pengujian ini terdiri dari pengujian mengelola *multitenancy*. Pada pengujian *multitenancy*, *tenant* baru yang akan mendaftarkan *tenant*-nya untuk dapat mengakses sistem memasukkan data berupa nama basis data yang akan dibuat. Apabila basis data berhasil dibuat, maka *tenant* dapat mulai mengakses dengan pengguna awal sebagai administrator. Rincian skenario pengujian pada fitur *multitenancy* dapat dilihat pada Tabel 5.10.

Tabel 5.18 Pengujian Fitur *Multitenancy*

ID	UJ.MTC
Referensi	Multitenancy
Nama	Mengelola Multitenancy
Tujuan Pengujian	Menguji kemampuan sistem dalam mengelola Multitenancy
Skenario 1	Pengguna mengakses Tenant
Kondisi Awal	Pengguna berada pada halaman <i>login</i>
Data Uji	Data <i>Tenant</i>

Langkah Pengujian	Pengguna memasukkan data uji ke form yang tersedia dan memilih tombol login
Hasil Yang Diharapkan	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Yang Didapat	Pengguna berada di halaman utama dengan data <i>tenant</i> -nya
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman utama



Gambar 5.11 Hasil Pengujian Multitenancy

5.3.4 Pengujian Fitur Mengelola Role Based Access Controll (RBAC)

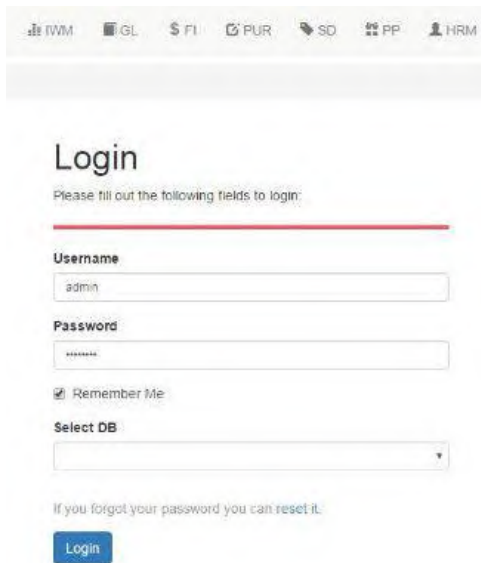
Pengujian ini terdiri dari pengujian mengelola *Role Based Access Control (RBAC)*. Rincian skenario pengujian pada fitur *Role Based Access Control (RBAC)* dapat dilihat pada Tabel 5.15.

Tabel 5.19. Pengujian fitur Role Based Access Control (RBAC)

ID	UJ.RBAC
Referensi	RBAC
Nama	Pengujian fitur <i>Role Based Access Control (RBAC)</i> .
Tujuan Pengujian	Menguji kemampuan aplikasi mengelola data <i>Role Based Access Control (RBAC)</i> .
Skenario 1	<i>Pengguna membuat user baru.</i>

Kondisi Awal	Pengguna berada pada halaman dashboard admin.
Data Uji	Inputan data <i>user</i> baru:
Langkah Pengujian	Pengguna memilih kegiatan <i>Add New User</i> , lalu memasukkan data inputan dan menekan tombol “ <i>Add User</i> ”.
Hasil Yang Diharapkan	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .
Hasil Yang Didapat	<i>User</i> yang baru terdapat pada halaman daftar <i>user</i> .
Hasil Pengujian	Berhasil
Kondisi Akhir	Pengguna berada pada halaman halaman daftar <i>user</i> dengan tambahan data <i>user</i> yang baru.
Skenario 2	<i>Pengguna menyunting data user tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>user</i> .
Data Uji	Perubahan data <i>user</i> :
Langkah Pengujian	Pengguna memilih kegiatan menyunting <i>user</i> tertentu, lalu melakukan perubahan data dan menekan tombol “ <i>Update</i> ”.
Hasil Yang Diharapkan	<i>User</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Yang Didapat	<i>User</i> yang disunting mengalami perubahan data sesuai inputan.
Hasil Pengujian	Berhasil.
Kondisi Akhir	Pengguna berada pada halaman daftar <i>user</i> .
Skenario 3	<i>Pengguna menghapus user tertentu.</i>
Kondisi Awal	Pengguna berada pada halaman daftar <i>user</i> .
Data Uji	-

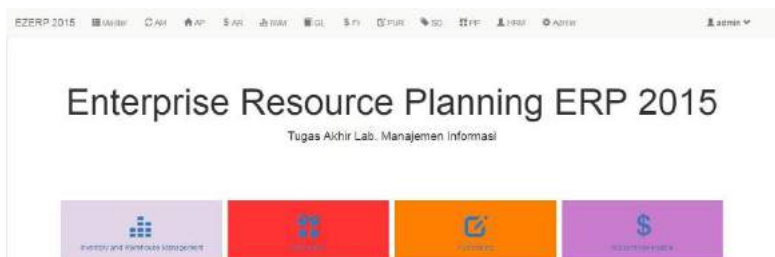
Hasil pengujian fitur RBAC dapat dilihat pada Gambar 5.1- Gambar 5.1.



The screenshot shows a web application interface with a top navigation bar containing icons and labels for various modules: IWM, GL, FI, PUR, SD, PP, and HRM. Below the navigation bar is a large, light gray rectangular area. In the center of this area is a 'Login' form. The form has the title 'Login' in a large, bold font. Below the title is a subtitle 'Please fill out the following fields to login:'. The form contains four input fields: 'Username' with the value 'admin', 'Password' with masked characters, a 'Remember Me' checkbox which is checked, and a 'Select DB' dropdown menu. Below the form is a link 'If you forgot your password you can reset it.' and a blue 'Login' button.

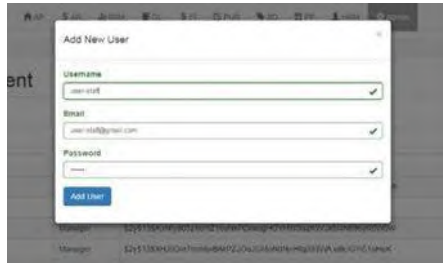
Gambar 5.12 Proses Login User Admin

Gambar 5.2 menunjukkan hasil pengujian untuk proses login oleh user admin.



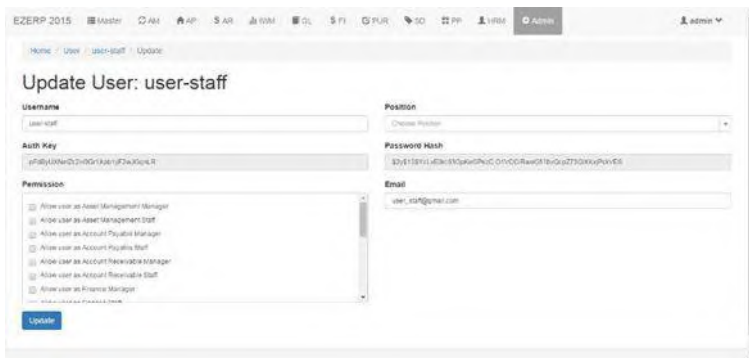
Gambar 5.13. Tampilan awal setelah login berhasil dilakukan

Gambar 5.14 menunjukkan hasil pengujian berupa tampilan awal setelah proses login oleh user admin.



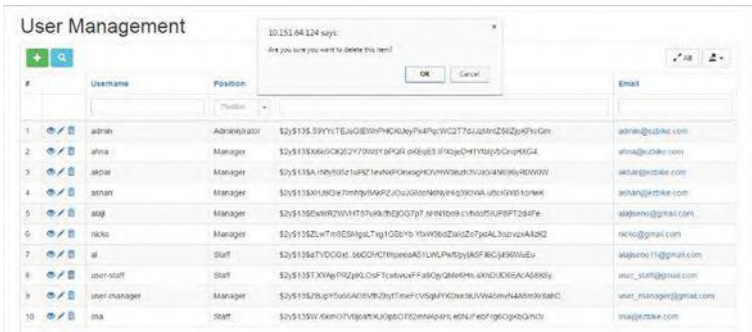
Gambar 5.14. Admin menambahkan user baru

Gambar 5.14 menunjukkan proses penginputan data *user* untuk skenario 1. Sedangkan untuk Gambar 5.15 menunjukkan hasil pengujiannya, dimana dapat dilihat penambahan pada halaman daftar user yang baru.



Gambar 5.15. Proses user admin menyunting data user

Gambar 5.15 menunjukkan proses penyuntingan data *user* untuk skenario 2.



Gambar 5.16. Proses user admin menghapus data user

Gambar 5.16 dapat dilihat proses penghapusan *user* untuk skenario 3. Dengan melihat hasil pengujian pada ketiga skenario diatas, bisa disimpulkan bahwa Kasus Penggunaan RBAC telah bekerja dengan baik seperti yang diharapkan.

5.4 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian kebutuhan fungsional dan evaluasi pengujian ketertarikan partisipan terhadap aplikasi.

5.4.1 Evaluasi Pengujian Fungsionalitas

Rangkuman mengenai hasil pengujian fungsionalitas dapat dilihat pada Tabel 5.16. Berdasarkan data pada tabel tersebut, semua skenario pengujian berhasil dan program berjalan dengan baik. Sehingga bisa ditarik kesimpulan bahwa fungsionalitas dari aplikasi telah dapat bekerja sesuai dengan yang diharapkan.

Tabel 5.20 Evaluasi Pengujian Fungsionalitas

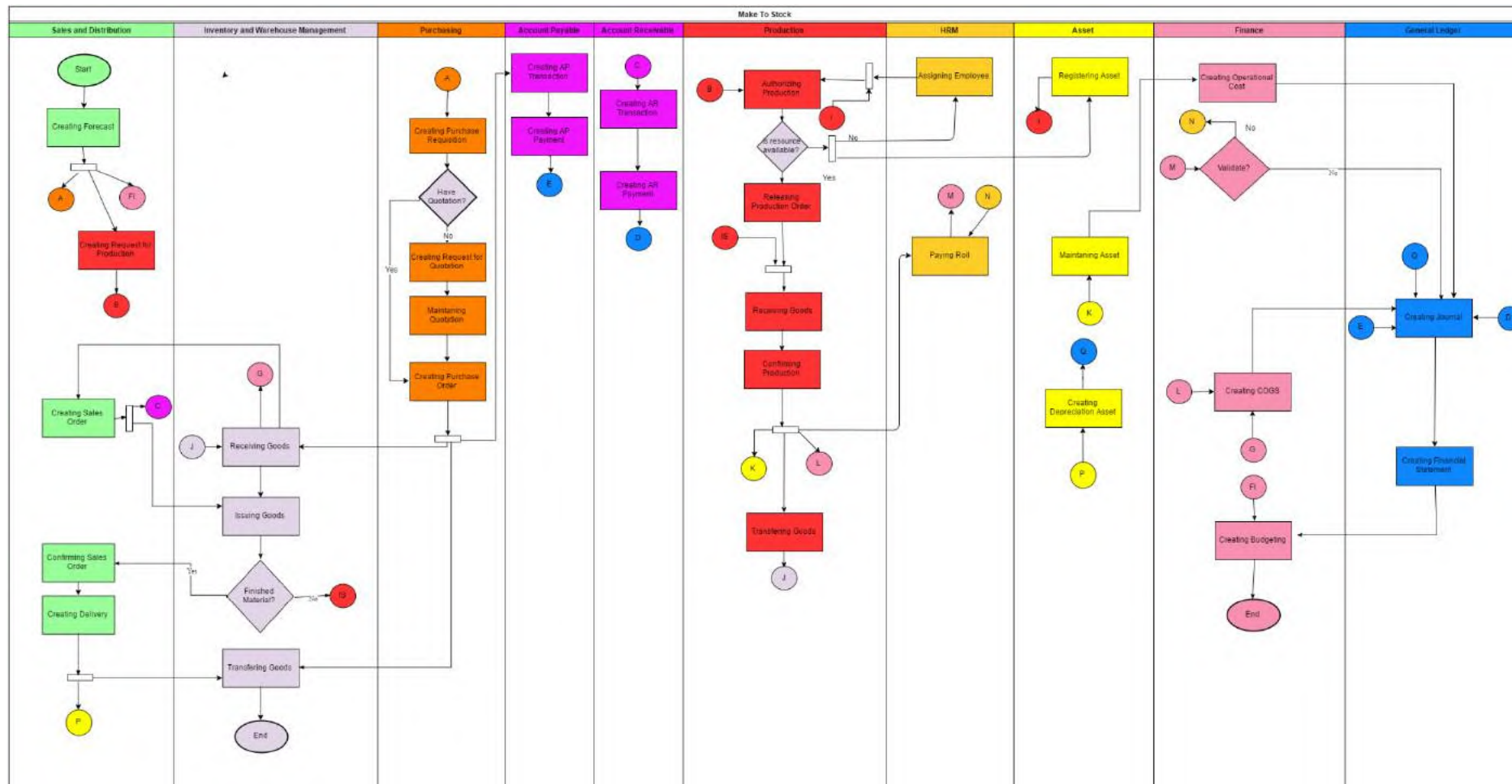
No.	ID	Nama Kasus Penggunaan	Skenario	Hasil
1	UJ.UC-001	Mengelola struktur perusahaan	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil

No.	ID	Nama Kasus Penggunaan	Skenario	Hasil
2	UJ.UC-002	Mengelola struktur kepegawaian	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
3	UJ.UC-003	Mengelola perekrutan pegawai	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
4	UJ.UC-004	Mengelola penempatan pegawai	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
5	UJ.UC-005	Mengelola training	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
6	UJ.UC-006	Mengelola bisnis goal	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
7	UJ.UC-007	Mengelola penggajian pegawai	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
8	UJ.UC-008	Mengelola pengurangan	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
9	UJ.UC-009	Melihat halaman Dashboard	Skenario 1	Berhasil
10	UJ.UC-010	Melihat laporan pegawai	Skenario 1	Berhasil
11	UJ.UC-011	Melihat data training	Skenario 1	Berhasil
12	UJ.UC-012	Melihat data pegawai	Skenario 1	Berhasil
13	UJ.RBAC	Mengelola RBAC	Skenario 1	Berhasil
			Skenario 2	Berhasil
			Skenario 3	Berhasil
14	UJ.MTC	Mengelola Multitenancy	Skenario 1	Berhasil
			Skenario 2	Berhasil

[Halaman ini sengaja dikosongkan]

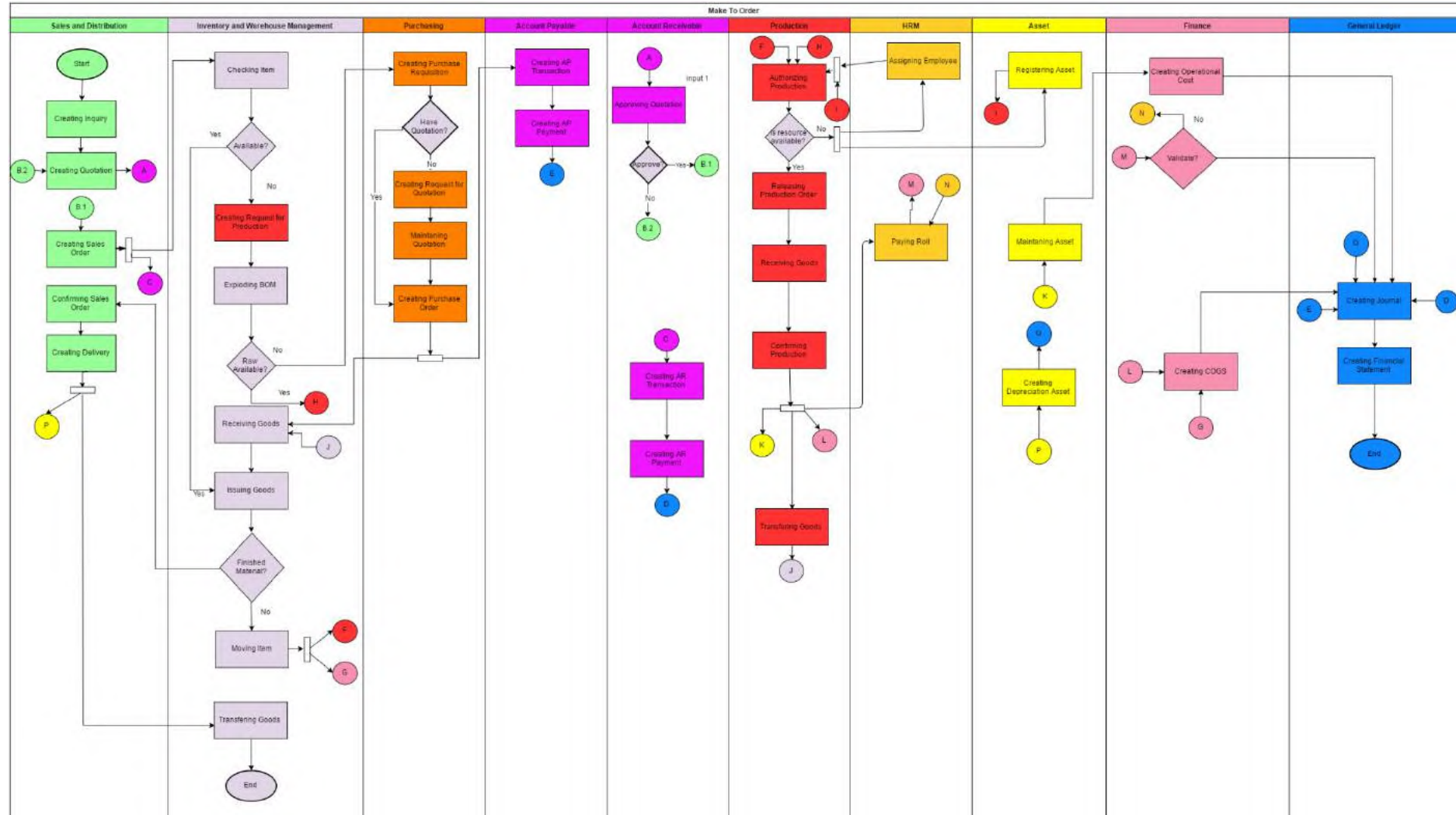
LAMPIRAN A – PROSES BISNIS

A.1. Proses Bisnis *Make To Stock* ERP 2016



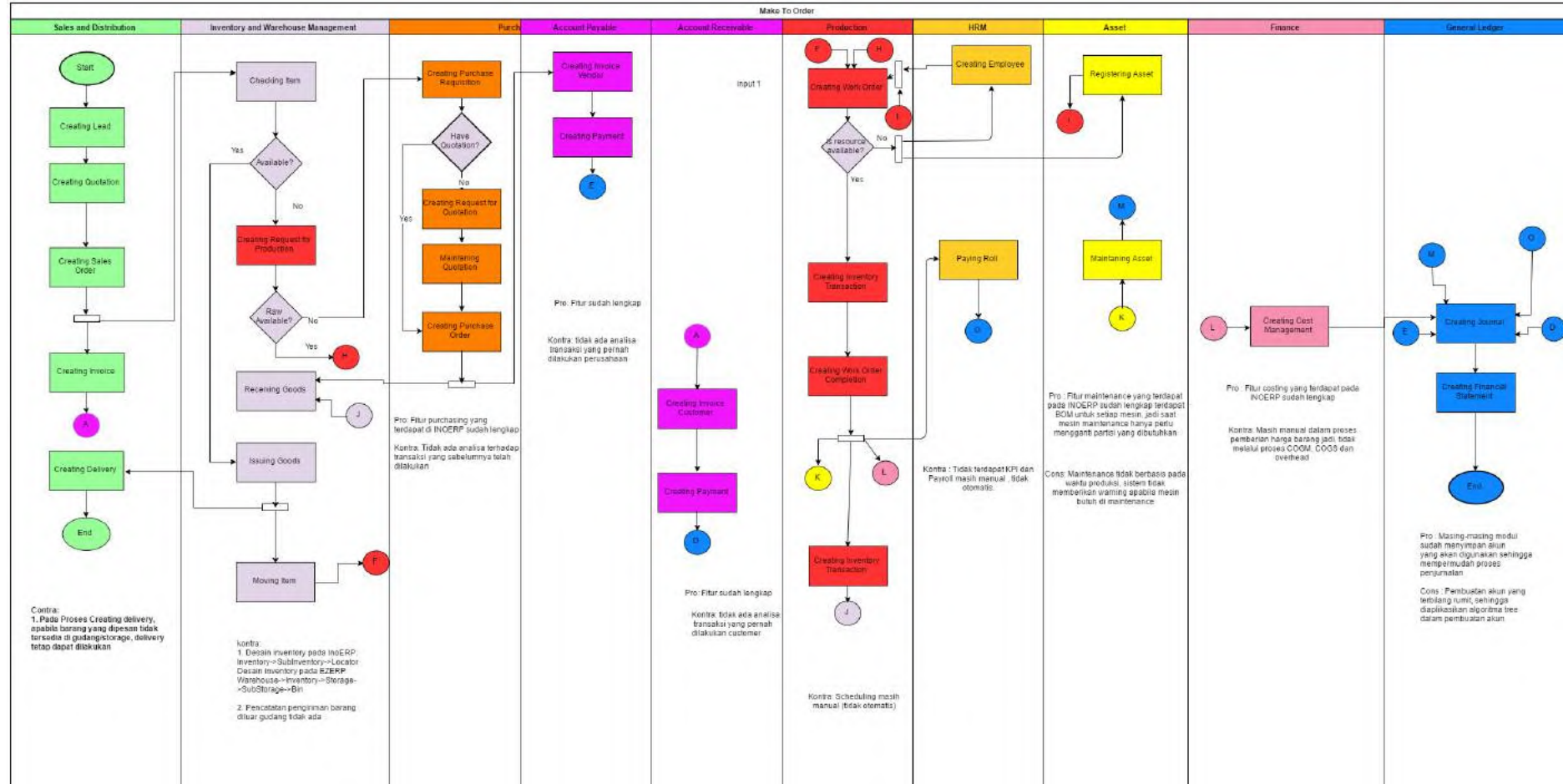
Gambar A.1 Workflow *Make To Stock* Level 1

A.2 Proses Bisnis Make To Order ERP 2016



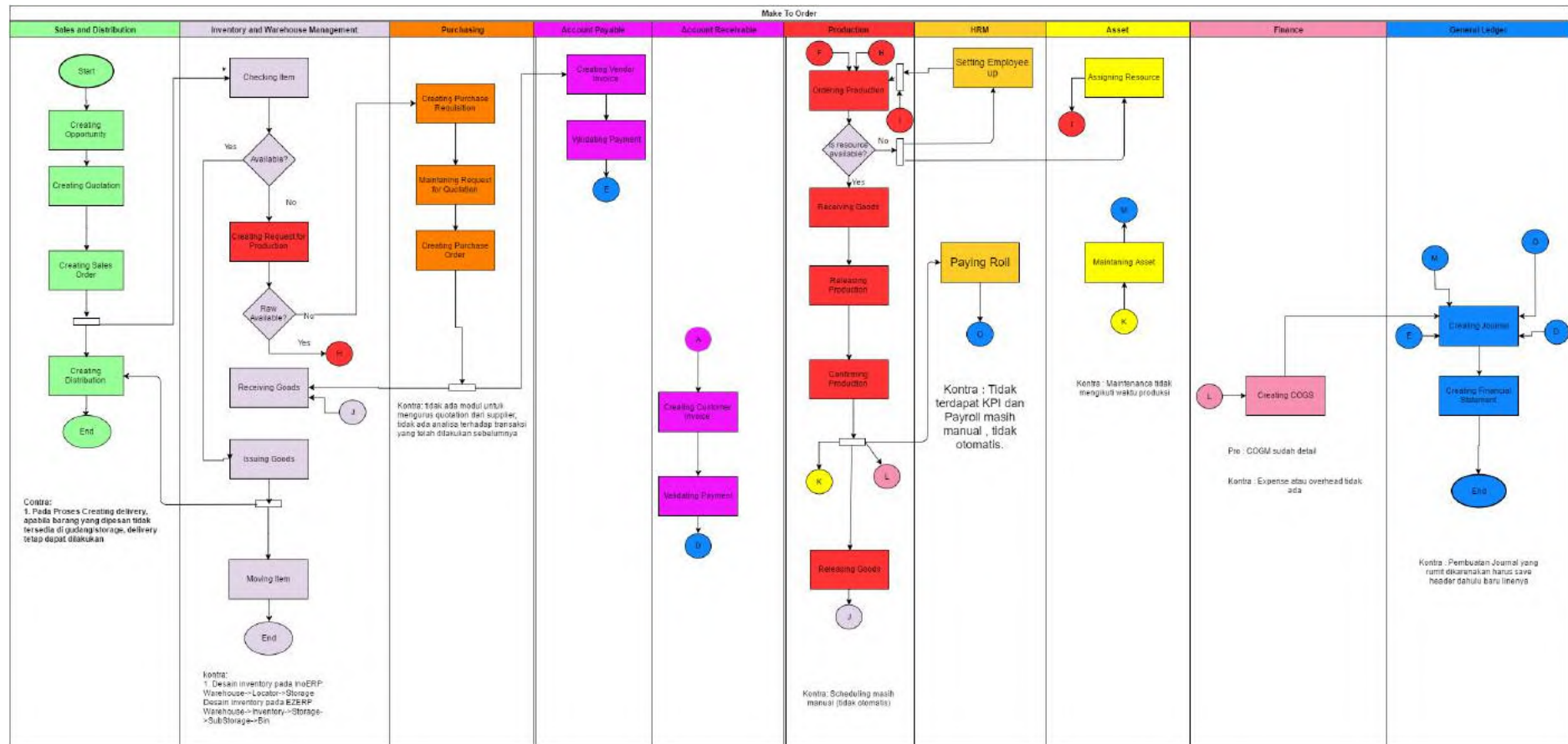
Gambar A.2 Workflow Make To Order Level 1

A.3 Proses Bisnis InoERP



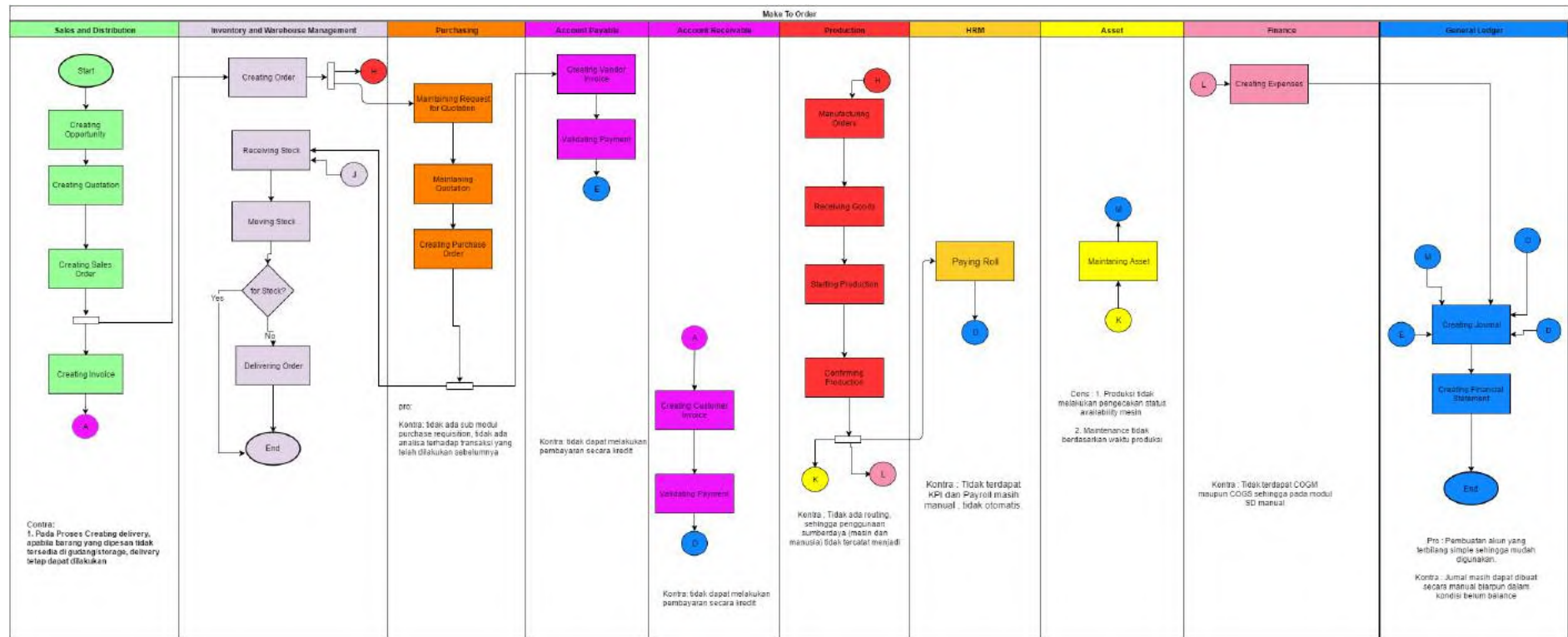
Gambar A.3 Workflow InoERP

A.4 Proses Bisnis Adempiere



Gambar A.4 Workflow Adempiere

A.5 Proses Bisnis Odoo



Gambar A.5 Workflow Odoo

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Pembuatan modul *Human Resource Management* (HRM) membantu mengelola sumber daya manusia yang ada di perusahaan. Mulai tahap perekrutan sampai tahap pengurangan pegawai. Perhitungan gaji secara menyeluruh sangat membantu perusahaan dalam menggaji pegawai berdasarkan daftar hadir pegawai.
2. Penggunaan arsitektur modularity pada aplikasi ERP ini dapat mengatasi masalah integrasi data. Pada arsitektur MVC normal, perbedaan antar modul sangat sulit untuk dikenali maupun digunakan. Maka dari itu, perlu adanya *modularity* yang di dalam arsitektur MVC dimana setiap modul memiliki MVC tersendiri dalam satu folder.
3. *Multitenancy* memungkinkan pengguna untuk bisa mengakses aplikasi yang sama akan tetapi dengan data dari masing-masing *tenant*, Dalam pengaplikasian *multitenancy* dibantu dengan adanya *Role Based Access Control* (RBAC) untuk dapat membedakan kepemilikan data.
4. Basis data terdistribusi digunakan sebagai solusi untuk dapat menyelesaikan masalah ketika salah satu *tenant* mengalami *fail-over* dan *tenant* lain tetap dapat mengakses aplikasi, dengan menggunakan replikasi dan fragmentasi yang disediakan oleh *MySQL Cluster*.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan:

1. Pengembangan aplikasi ERP dapat mengikuti aspek Internet of Things dimana data-data internal maupun eksternal perusahaan dapat diakses. Hal ini akan sangat membantu perusahaan dalam mengambil keputusan.
2. Penambahan modul sangat memungkinkan pada ERP ini. Dimana modul yang ditambahkan dapat terintegrasi dengan modul-modul yang lain.

DAFTAR PUSTAKA

- [1] A. V. Alloserung, Rancang Bangun Aplikasi Berorientasi Arsitektur Service (SOA) dengan Pendekatan Workflow pada Domain Customer Relationship Management (CRM) untuk Sistem Enterprise Resource Planning, Surabaya: Institut Teknologi Sepuluh Nopember, 2013.
- [2] S. R. Magal and J. Word, INTEGRATED BUSINESS PROCESSES with ERP SYSTEMS, JOHN WILEY & SON S, I N C.
- [3] S. P. Robbins and M. Coulter, Manajemen, Jakarta: PT INDEKS, 2009.
- [4] T. Connolly and C. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, Harlow: Pearson Education, 2005.
- [5] M. T. Özsu and P. Valduriez, Principles of Distributed Database Systems, New York: Springer, 2011.
- [6] R. Wolter, G. Carraro and F. Chong, "Multi-Tenant Data Architecture," Microsoft Corporation, June 2006. [Online]. Available: <https://msdn.microsoft.com/en-us/library/aa479086.aspx>. [Accessed December 2015].
- [7] S. David, J. Jain and K. Jain, "Study of Human Resource Information System in Global Milieu," *International Journal of Management and Social Sciences Research*, vol. 2, no. 11, pp. 60-62, 2013.
- [8] R. S and B. M, "Structure of Human Capital Enhancing Human Resource Management Practices in India," *International Journal of Business and Management*, vol. 4, no. 5, p. 13, 2009.
- [9] F. Rinaldi, "Menghitung Pajak Penghasilan & Tarif Pph 21 Terbaru 2016," Kembar.pro, 2016. [Online]. Available: <http://www.kembar.pro/2015/10/menghitung-pajak->

[penghasilan-tarif-pph-21-terbaru-2015.html](#). [Accessed 23 April 2016].

- [10] G. Coulouris, J. Dollimore, T. Kindberg and G. Blair, Distributed Systems, Boston: Pearson Education, 2012.

BIODATA PENULIS



Penulis, **Dwi Al Aji Suseno**, lahir di Banyuwangi, 19 Juni 1993. Penulis menempuh pendidikan sekolah menengah pertama di SMP Negeri 1 Banyuwangi dan penulis mendapatkan beasiswa selama SMA di SMA Negeri 10 Malang (Sampoerna Academy). Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Fakultas Teknologi dan

Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif menjadi administrator Laboratorium Manajemen Informasi Teknik Informatika dan aktif dalam organisasi tingkat jurusan, fakultas.

Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Manajemen Informasi (MI) dan memiliki ketertarikan di bidang *Web* dan *Mobile Application Development*. Penulis dapat dihubungi melalui email: alajiseno@gmail.com.